UNIVERSITÀ DEGLI STUDI DELL'AQUILA
DEPARTMENT OF INFORMATION ENGINEERING AND COMPUTER SCIENCE AND MATHEMATICS

DOCTORAL THESIS

Machine Learning Approaches for Medical Data Understanding and Therapy Recommendation in Alignment with FAIR Principles

Candidate:
Payel PATRA

Advisor:
Prof. Antinisca DI MARCO

Co-Advisors:
Daniele DI POMPEO

Doctoral Program Coordinator:
Prof. Davide DI RUSCIO

A thesis submitted in fulfillment of the requirements for the degree of Doctor of Philosophy in

Information and Communication Technology
Curriculum: Emerging computing models, software architectures, and intelligent systems

XXXVIII Cycle

SSD INFO-01/A

A.Y. 2024/2025

Abstract

Payel PATRA

Machine Learning Approaches for Medical Data Understanding and Therapy Recommendation in Alignment with FAIR Principles

Medical Data Analysis (MDA) plays an important role in improving healthcare outcomes by converting complex clinical data into meaningful and actionable insights. In this thesis, we enhance this goal by integrating state-of-the-art AI approaches [1]. In addition, this thesis adheres to the FAIR guidelines (i.e., findable, accessible, interoperable, and reusable [2]), aligned with the general perspective of Open Science [3]. FAIR principles compliance aims to promote transparent and interoperable workflow procedures that support a collaborative environment and increase community trust in biomedical studies.

In particular, in Part I of this thesis, we developed an AI-based pipeline to analyze the raw clinical records for extracting structured knowledge from unstructured clinical notes in the form of key clinical entities. Domain-specific BERT-based models—such as *BioBERT, BioClinicalBERT, BlueBERT, RoBERTa, and PubMedBERT*—are used to identify important entities, including *Age, Gender, Diseases, Symptoms, Medications, Dosage, and Disease Stage*[1]. The extracted entities are subsequently organized into medical knowledge graphs that capture clinically meaningful relationships among patients and serve as machine-actionable, FAIR-compliant data resources. Building on these graph-based representations, we developed a Reinforcement Learning–Guided Medicine Optimization (RLGMO) pipeline that uses Graph Neural Networks (GNNs) and Reinforcement Learning (RL) to model and learn the relationships between patients, diseases, and treatment options.

In PART II of this thesis, twenty-two FAIR assessment (FAIR-a) tools are systematically evaluated to identify effective solutions for their integration into Open Science platforms (OSPs). Each tool is evaluated based on some important criteria, including technical performance, level of automation, metadata validation, interoperability, usability, and runtime efficiency. The analysis provides guidance for selecting and improving FAIR-enabling technologies to support ethical data governance and reproducible research by highlighting the strengths and limitations of existing FAIR assessment tools, such as inconsistent identifier handling and low levels of automation.

Taken together, these contributions ensure responsible and transparent data management by aligning medical data analysis with FAIR-oriented Open Science practices. The structured clinical outputs generated by our MetaGRO (*Medical Entity and Text Analysis for Graph-based Reinforcement Optimization*) framework are integrated into the *SoBigData Research Infrastructure (RI)*— a European D4Science-based research infrastructure for FAIR multidisciplinary data mining and analytics [4]. This integration enables intelligent, reusable, and ethically sound biomedical workflows that foster openness, collaboration, and innovation in contemporary medical research.

[1] In this thesis, we focused on cancer disease.

Acknowledgements

I would like to express my heartfelt appreciation to my supervisor, Prof. Antinisca Di Marco, for her constant encouragement, invaluable guidance, and inspiring mentorship throughout this research journey. Her insightful feedback, patience, and support have profoundly shaped both my academic and personal growth, motivating me to become a more confident and independent researcher.

I am sincerely grateful to the University of L'Aquila's Department of Computer Science faculty and staff for providing a stimulating academic environment and technical assistance that greatly facilitated my work. My heartfelt thanks also go to my professors, Prof. Antinisca Di Marco, Prof. Thanh Phuong Nguyen, Prof. Daniele Di Ruscio, and Prof. Vittorio Cortellesa, for their invaluable guidance and encouragement. I would especially like to thank Dr. Daniele Di Pompeo for his constructive feedback, technical insights, and thoughtful discussions, whose guidance and mentorship significantly enriched this research and played an important role in my development during the PhD period. I am also thankful to Michele Tucci, whose aggregate knowledge, kindness, and assistance helped with the accomplishment of my PhD.

I am deeply thankful to my friends and colleagues who shared this academic journey with me. Their companionship, motivation, and countless memorable moments brought light and balance to my PhD life. I would also like to thank my friends Andrea D., Gennaro, Alina, Andrea M., Mashal, Federico, Riccardo, Roberta, Mario, and Isabella for their support and companionship throughout this journey, and all those who made this experience more enjoyable. I would also like to thank Giordano D'Aloisio and Andrea Bianchi for their constant support, collaboration, and encouragement throughout this journey, which made both the research and the experience much more enjoyable.

My sincere gratitude goes to my parents and sister, whose unconditional love, patience, and belief in me have been my greatest source of strength. Their constant support and encouragement gave me the courage and resilience to overcome challenges and reach this milestone. I am also deeply thankful to my extended family members for their care and encouragement, which have meant a great deal to me throughout this journey.

Finally, I would like to thank all the funding organizations, research initiatives, and institutional resources that made this work possible.

Contents

List of Figures

List of Tables

List of Algorithms

To my parents, whose
unwavering love and sacrifices
shaped the person I am; to my
sister, my quiet strength and
constant support; and to my
respected professors, whose
guidance, encouragement, and
wisdom have inspired me
throughout this journey.

Chapter 1

Introduction

MEDICAL data analysis (MDA) has emerged as one of the most significant areas of modern scientific research, driving improved diagnosis, patient-centered clinical decision-making, and data-driven healthcare innovation. It involves interpreting large-scale information from diverse sources—such as electronic health records (EHRs) [5], clinical notes [6], and laboratory results [7]—to uncover meaningful insights. With the growing use of artificial intelligence (AI) [1] in healthcare, automated systems have emerged to process, interpret, and reason over complex biomedical data. These systems are transforming clinical data analysis and decision-making processes.

Natural Language Processing (NLP) [8] is a well-known technique for comprehending clinical free-text data among the current AI technologies [1]. This helps convert diagnostic (such as radiologic) reports, patient discharge data, and free-text clinical information handwritten by physicians into a format that a machine can process further. Transformer-based language models such as BERT and its healthcare-specific variants are widely used in current NLP studies to improve the understanding of clinical terminology and applications. Clinical Named Entity Recognition (NER) [9] or *medical entity extraction* [10] are two major technical issues under this domain. The entities in this scenario are categorized into clinical features such as *diseases, symptoms, drugs, dose, and other aspects related to the therapies.*

The field of biomedical text analysis [11] has evolved from early rule-based and ontology-driven systems to transformer-based deep learning models [12]. Recent domain-specific models such as BioBERT[1], BioClinicalBERT[2], BlueBERT[3], and PubMedBERT[4] have achieved state-of-the-art performance in biomedical information extraction, while RoBERTa[5], a general-purpose language model, is often used as a baseline (as in this study) for comparative evaluation. These transformer-based models capture contextual dependencies across medical text, improving entity recognition accuracy and robustness. Their effectiveness has enabled the development of intelligent clinical systems that support diagnostic predictions, therapeutic recommendations, and the interpretation of real-time data.

Our research study introduces the MetaGRO *(Medical Entity and Text Analysis for Graph-based Reinforcement Optimization)* Framework, which forms the core contribution of this thesis. The framework brings together several domain-specific transformer-based BERT models for medical entity extraction, *knowledge graph* (KG) construction, and reinforcement learning–based optimization for therapy recommendation.

[1] `https://huggingface.co/dmis-lab/biobert-v1.1`
[2] `https://huggingface.co/emilyalsentzer/Bio_ClinicalBERT`
[3] `https://huggingface.co/bionlp/bluebert_pubmed_uncased_L-24_H-1024_A-16`
[4] `https://huggingface.co/NeuML/pubmedbert-base-embeddings`
[5] `https://huggingface.co/docs/transformers/en/model_doc/roberta`

Within MetaGRO, the extracted entities—such as diseases, medications, and symptoms—are transformed into structured graphs that capture semantic relationships across patients (RLGMO), representing the final module of our framework. Furthermore, *Graph Neural Networks* (GNNs) [13] and the *Reinforcement Learning–Guided Medicine Optimization* (RLGMO) module are integrated to enable pattern discovery (e.g., identifying relationships such as an RLGMO_ID associated with a disease or symptom that leads to an appropriate medication), drug discovery, and personalized treatment suggestions. For example, the model can identify patterns where a symptom such as chest pain frequently co-occurs with cardiovascular diagnoses, or where a specific diagnosis consistently leads to a preferred medication choice. By converting unstructured clinical text into interpretable knowledge representations, the MetaGRO framework bridges data understanding and clinical decision-making in a transparent, reproducible, and intelligent manner.

For achieving a broader impact in biomedical data research and advancing open science, it is equally essential to maintain interoperability, transparency, and reproducibility in AI-driven analysis. In this context, the FAIR principle [2] plays a crucial role. These principles are not new; rather, they have long guided the management of research data since being introduced by Wilkinson et al. in 2016 [14]. FAIR—standing for *Findable, Accessible, Interoperable, and Reusable*—provides a structured and sustainable framework to ensure that data can be efficiently discovered, shared, and reused by both humans and machines. Within the biomedical domain, where clinical data [15] are inherently complex, heterogeneous, and sensitive, considering the FAIR principles strengthens data governance [16] and supports ethical compliance [17]. Applying FAIR principles in medical data management [2], [15] strengthens data governance and supports secure, interoperable research collaboration.

The intersection of AI [1] and FAIR data management [2] represents a pivotal advancement in biomedical informatics. AI models enable intelligent interpretation of medical data, while FAIR principles ensure transparency, accountability, and long-term reusability of research outputs. Together, AI-driven analytics and FAIR-aligned governance enable transparent and reproducible healthcare innovation. In this thesis, this connection is realized through the MetaGRO framework, which generates structured clinical knowledge from unstructured medical text while ensuring that the resulting datasets, models, and analytical outputs can be stored and shared in a FAIR-compliant research infrastructure.

In this respect, our work presents an AI-driven framework for clinical data analysis, and the structured outputs generated by our pipeline's final module, RLGMO, are deposited into the *SoBigData Research Infrastructure (RI)*[6] [18], enabling FAIR-aligned sharing and reuse. Rather than developing a new Open Science Platform, this work employs an existing European research infrastructure that already supports open science practices and FAIR-compliant data dissemination across the research community. The selection of SoBigData RI is motivated by its alignment with European open science initiatives and its capability to support responsible data sharing within a collaborative research environment. Although SoBigData RI

[6]SoBigData RI is a long-term European research infrastructure supporting responsible data science, FAIR/FACT principles, and cross-national collaboration. Funding acknowledgment: European Union — NextGenerationEU — National Recovery and Resilience Plan (Piano Nazionale di Ripresa e Resilienza, PNRR) — Project: "SoBigData.it — Strengthening the Italian RI for Social Mining and Big Data Analytics" — Prot. IR0000013 — Avviso n. 3264 del 28/12/2021.

is not specifically designed as a biomedical-focused OSP, it provides relevant functionalities for managing and disseminating structured research outputs in a FAIR-compliant manner, making it a suitable platform for supporting the reuse and accessibility of the data produced in this work.

Rather than embedding FAIR principles directly within the AI model—an acknowledged challenge in biomedical research [19]—our framework achieves FAIR compliance through the responsible sharing and accessibility of processed clinical information via the SoBigData platform. This ensures that the resulting data are transparent, reusable, and interpretable through machine intelligence, thereby contributing to sustainable and reproducible global health research. The following sections describe the contributions of this thesis, the identified challenges, and the strategies employed to address them (please refer to Section 1.1); list the publications on which this thesis is grounded (Section 1.2), and outline the overall structure of the thesis (Section 1.3).

1.1 Thesis Contribution

The main contribution of this thesis is the development of an integrated framework, named the MetaGRO *(Medical Entity and Text Analysis for Graph-based Reinforcement Optimization)* framework, which combines artificial intelligence and reinforcement learning to advance medical data analysis [20] and support clinical decision-making. The framework aligns with FAIR data practices [21] through responsible management, curation, and sharing of processed outputs. Figure 1.1 summarizes the methodological workflow of MetaGRO and shows how each part of the framework operates. The solid arrows represent the step-by-step flow of data, beginning with the collection of unstructured clinical text. These texts are cleaned, pre-processed, and then passed to a BERT-based clinical entity extraction method to identify diseases, symptoms, medications, and cancer stages [15]. The complete workflow for preprocessing and entity extraction is described in Chapter 4. The extracted entities are then converted into knowledge graphs that capture medical relationships based on standardized ontologies[7] and support advanced reasoning over clinical data.

The light blue rectangular boxes inside the MetaGRO framework in this figure show the main data forms produced at each stage, including processed text, extracted entities, knowledge graphs, and the final therapy recommendations generated by the RLGMO (*Reinforcement Learning Guided Medicine Optimization*) model. The construction of the knowledge graphs, pattern detection using Graph Neural Networks (GNNs), and the functioning of RLGMO are explained in Chapter 5. The dashed boxes in the figure indicate the thesis chapters where each component is covered in detail, helping the reader relate the visual workflow to the written methodology. The grey boxes represent supporting elements—such as open-science platforms and FAIR-assessment tools—that are not part of the computational pipeline but enable responsible data sharing, transparency, and reuse. The vertical arrow from the MetaGRO outputs to the grey region shows how anonymized graphs, RLGMO results, and other structured outputs are deposited into external research infrastructures for FAIR-compliant access and long-term reuse. These outputs—including

[7] Medical ontologies are standardized vocabularies that define medical terms and their relationships—such as diseases, symptoms, procedures, drugs, and clinical findings—to support consistent interpretation and interoperability across healthcare systems. Examples include UMLS, SNOMED CT, MeSH, ICD, and RxNorm.

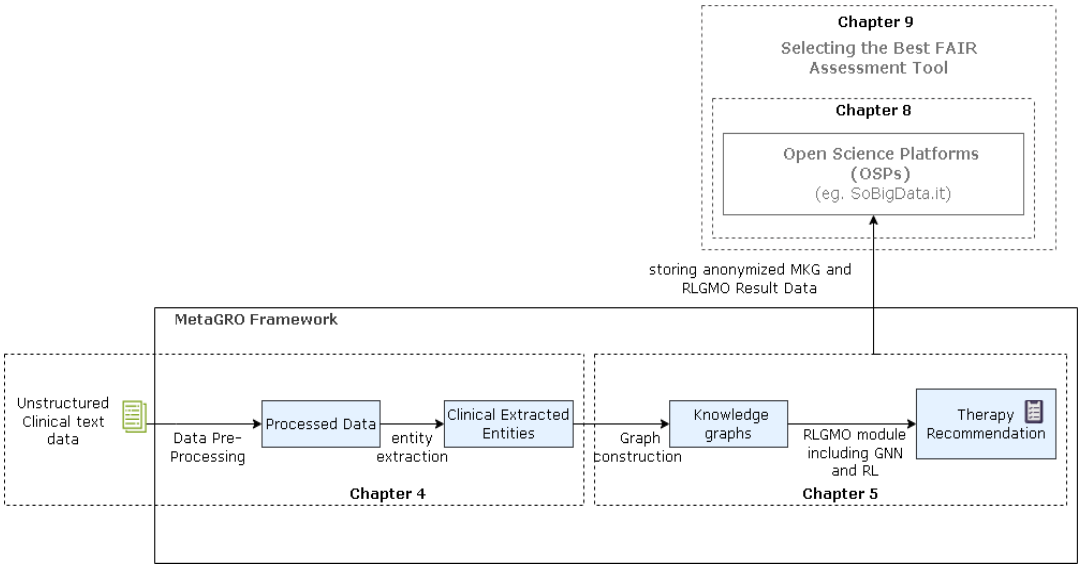


FIGURE 1.1: General workflow of the thesis contribution through the **MetaGRO** framework: *Medical entity and text analysis for graph-based reinforcement optimization (RLGMO — Reinforcement Learning Guided Medicine Optimization), leading to therapy recommendation.*

structured datasets, trained model files, graph embeddings, and evaluation results—collectively capture the analytical and predictive capabilities of MetaGRO.

These outputs are ultimately integrated into the *SoBigData.it* Open Science Platform, which provides a secure and interoperable environment for managing FAIR-ready clinical and medical data. Although these aspects are not explicitly shown in the figure, Chapter 7 provides a detailed explanation of the concept of open science, the purpose of data repositories, and why it is important to store research data in well-maintained and accessible repositories. The chapter also discusses how applying FAIR principles can improve the quality, reuse, and visibility of data within these repositories. Chapter 8 then focuses on assessing different FAIR-assessment tools and compares their performance to understand how effectively they support FAIR data evaluation. By depositing the generated datasets and related research artifacts—such as trained models, graph embeddings, and evaluation outputs—within a suitable open-science repository, the research process ensures alignment with FAIR principles and supports reproducibility, transparency, and long-term scientific reuse.

In this way, the MetaGRO framework provides a complete end-to-end workflow. It transforms unstructured clinical text into structured medical knowledge graphs, generates intelligent therapy recommendations through the RLGMO model, and ensures long-term storage of the resulting artifacts within a FAIR-compliant repository. Although FAIR principles mainly apply to data governance rather than to the internal design of algorithms, the outputs produced by MetaGRO are structured and machine-readable, enabling their integration into FAIR-compliant Open Science Platforms (OSPs). The FAIR principles [16] therefore guide the final stage of this process by ensuring that the results generated by MetaGRO are managed, shared, and reused in a transparent and sustainable manner within the broader open-science community.

1.1.1 Identified Challenges (CHs)

Although considerable research has focused on enhancing the efficiency, transparency, and FAIRness of data-driven healthcare systems, several key challenges still persist. These include the need for adaptive systems capable of providing improved therapy recommendations, the difficulty of converting unstructured clinical text into clear and well-organized medical information [15], and the absence of widely accepted standards for representing clinical data in a consistent and interoperable manner. Another challenge is the absence of proper and widely accepted methods to check and evaluate many FAIR-a tools [22] altogether, which limits a fair comparison of their performances.

In this thesis, these challenges are carefully studied and addressed through the development of an integrated research framework called the MetaGRO Framework *(Medical Entity and Text Analysis for Graph-based Reinforcement Optimization)*. This framework combines Artificial Intelligence (AI) [1], Knowledge Graph Modeling (KGM) [23], and Reinforcement Learning (RL)-based decision support [24]. The proposed framework aligns with the FAIR principles by generating structured outputs and depositing them within an open science platform (OSP) to support transparency, interoperability, and long-term reuse.

Part I – Challenges in Medical Data Analysis and Clinical Text Understanding

The initial part of this study focuses on the processing and structuring of clinical texts using AI-based clinical data analysis, generating organized medical key information [15]. These outputs serve as the foundation for later modules of our proposed framework MetaGRO, where they are prepared for effective use in downstream clinical and analytical tasks. Despite advances in biomedical informatics, challenges still remain in applying this knowledge to clinical decision-making and in converting raw medical text into structured and reusable clinical information.

CH$_1$ *Clinical data are highly unstructured, narrative-based, and difficult to process in a reusable computational format.*

Although clinical notes, electronic health records, and case histories contain important medical information, they are usually written as free-text narratives. This lack of structure makes the data difficult for machines to process and limits its usefulness for analysis, decision support, and further research. In addressing this challenge, the main research question that guided this part of the thesis was:

> **"How effectively can domain-specific transformer-based BERT models extract medical entities from clinical text, and which model-evaluation strategy is most suitable for clinical use?"**

To answer this question in a systematic way, the main research question is divided into three sub-questions, which are presented and analyzed in detail in Chapter 4. These sub-questions help evaluate the extraction capability of BERT-based models, identify appropriate evaluation criteria, and determine the most suitable model for downstream clinical applications.

CH$_2$ *Iterative and intelligent methods for therapy or medication recommendation are limited in current AI-driven systems [1].*

Most existing computational approaches for therapy optimization operate in a static manner, relying on fixed datasets or predefined models that do not adapt to evolving clinical information. Consequently, they lack the capacity to refine recommendations in response to changing patient conditions or treatment outcomes. In contrast, feedback-driven reinforcement learning enables continuous policy improvement, supporting dynamic and personalized therapeutic decision-making. In this context, the main research question that guided this part of the thesis was:

> **"How can knowledge graphs and reinforcement learning together help discover clinical patterns and improve personalized therapy recommendations in oncology?"**

To explore this question in a structured way, three sub-questions were defined, and these are addressed in detail in Chapter 5.

Part II – Challenges in Advancing FAIR-Aligned Practices within Open Science Platforms

After developing the integrated MetaGRO Framework and its intelligent RL-GMO module in the first part of this study, the second part focuses on identifying suitable platforms that can effectively promote and share FAIR-aligned knowledge or information within the Open Science environment by evaluating different FAIR assessment tools [22]. Selecting the most appropriate and user-friendly platforms to ensure data reusability and transparency still remains a challenge, even though the FAIR movement is now widely accepted in the research community. When working with medical and clinical data, this challenge becomes even more important because such data cannot always be made publicly available due to privacy rules and ethical concerns. Therefore, this research also explores open science platforms that can support FAIR data sharing in a responsible and secure way while protecting patient confidentiality.

CH$_3$ *Selecting an appropriate open science platform (OSP) to effectively promote FAIRness and support ethical and secure data sharing.*

Selecting an appropriate repository for the sharing of FAIR data is a challenging procedure because different repositories have different policies, safeguarding mechanisms, compatibility, and accessibility. When choosing a repository, the problem of privacy in biological data comes into play because private patient data cannot be publicly shared. As a result, choosing the repository depends on achieving a balance between sharing FAIR data and maintaining privacy.

In this context, the guiding question for this part of the work was:

> **"Which open science platform is best to support FAIR principles while ensuring secure data sharing in the biomedical research domain?"**

The comparison between open science repositories, along with the reasoning behind the selected platform, is presented in detail in Chapter 7.

CH$_4$ *Existing FAIR assessment tools exhibit technical and functional limitations that impact evaluation reliability.*

Although numerous FAIR assessment tools have been developed [22], they exhibit limitations such as restricted automation, non-standardized scoring mechanisms, and limited interoperability. These constraints reduce the reliability and comparability of FAIR evaluations across different research domains.

In this context, the main research question that guided this part of the work was:

> **"To what extent can Open Science Platforms (OSPs) implement the FAIR principles, and how effective are FAIR-assessment tools in evaluating this implementation and guiding improvements in FAIRness and usability?"**

This question is explored in detail through four sub-questions, which are discussed in Chapter 8 .

1.1.2 Proposed Contributions (CNs)

In order to address the concerns raised, this thesis presents a comprehensive approach called the MetaGRO Framework, which incorporates the most recent developments in the use of AI for medical data analysis [1]. Each section of this thesis is based on its pertinent research challenges, and the suggested contributions include a seamless connection between ethical data stewardship and intelligent clinical decision assistance.

Part I – Contribution in Medical Data Analysis and Clinical Text Understanding

The first section of this research focuses on the technological aspects of the MetaGRO Framework, which include AI-based medical data analysis [1], such as clinical entity recognition, medical knowledge graph creation, and individualized therapeutics based on reinforcement learning. Each of them aids in the conversion of unstructured clinical narratives generated during healthcare operations into comprehensible, organized, and data-driven healthcare outcomes.

CN$_1$ *Evaluation and Comparison of Domain-Specific Transformer Models for Medical Entity Extraction.*

In this study, transformer-based models, which include BioBERT, BioClinicalBERT, BlueBERT, RoBERTa, and PubMedBERT, will be assessed in terms of extracting structured information from unstructured clinician notes. Also, pretrained and fine-tuned transformer models will be considered to determine their effectiveness in extracting major medical information like diseases, symptoms, and medications.

CN$_2$ *Design and implementation of the Reinforcement Learning–Guided Medicine Optimization (RLGMO) model integrated within the MetaGRO Framework.*

This study proposes an RLGMO model, which represents the intelligent learning component of MetaGRO, to address the drawbacks of conventional decision-making systems. By using graph neural networks and reinforcement learning to medical knowledge graphs, RLGMO is able to offer dynamic and customized patient treatment suggestions. As a result, RLGMO has a major effect on how well treatment recommendations can be interpreted.

Part II - Contribution in Advancing FAIR-aligned Practices within Open Science Platforms

The second part of this research presents a comprehensive framework for evaluating FAIR-enabling tools and identifying suitable open science platforms for ethical and interoperable biomedical data management. In particular, this thesis introduces a classification framework that systematically assesses existing FAIR-a tools [22] to derive key evaluation dimensions for platform assessment. These dimensions enable a structured analysis of technical performance, usability, functionality, and runtime characteristics of the tools. Through this evaluation, the framework supports the identification and selection of appropriate open science platforms for FAIR-compliant biomedical data sharing within the European research ecosystem, while also revealing the strengths and limitations of existing FAIR-a tools.

CN$_3$ *Identifying and selecting a suitable open science platform for promoting FAIR-aligned research data.*

The second key contribution of this part is the analysis and selection of an appropriate platform to support the publication and sharing of FAIR-compliant research data. For this purpose, the SoBigData Research Infrastructure—an EU-funded platform [25] that provides tools, datasets, and guidelines for responsible data sharing—is considered a suitable option for managing and disseminating biomedical datasets. Using this platform ensures that the research data and results produced in this study can be stored, accessed, and reused in a transparent and sustainable manner over time.

CN$_4$ *Evaluation and enhancement of existing FAIR assessment tools to strengthen technical robustness.*

The study evaluates existing FAIR-a tools to identify gaps related to automation, metadata validation, and interoperability. The insights derived from this analysis contribute to improving FAIR-enabling technologies, supporting the development of more technically consistent and scalable solutions for diverse research contexts.

The principles of Open Science [3] are considered in this work to guide both the AI-based medical data analysis modules of the MetaGRO Framework and the FAIR-a tools evaluation. These elements are designed to align with the practices encouraged by the SoBigData Research Infrastructure (RI), a European initiative that supports transparent and multidisciplinary data sharing. SoBigData, which operates on the D4Science platform [4], provides a secure and collaborative environment for managing and analyzing the research data. Making the result of the outputs of our proposed framework compatible with the SoBigData ecosystem can be helpful for future researchers to adapt, reuse, and extend these resources, which can contribute to ongoing developments in biomedical data management and AI-driven clinical decision support.

While FAIR alignment is first and foremost a matter of responsible storage and dissemination infrastructure, issues of interoperability also play a role in the design of graph outputs and machine-readable representations in MetaGRO. This is to ensure that the framework is not only analytically sound but also interoperable with sustainable research infrastructures.

1.2 List of Publications

Every contribution in this thesis is derived from works that have been submitted to journals or scientific conferences. In chronological order, the full list of conference and workshop publications is shown below.

Conference Papers

C1. Patra, P., Di Pompeo, D., & Di Marco, A. (2025). *An Evaluation Framework for the FAIR Assessment tools in Open Science. arXiv preprint arXiv:2503.15929.*[8] Here in this paper, we applied the framework containing 22 FAIR-a tools, identified their strengths and gaps, and discussed how this evaluation can guide improvements in the tool's design and FAIR adoption.

C2. Patra, Payel, Antinisca Di Marco, and Phuong Than Nguyen. *"Entity Extraction in Clinical Summary of Mammary Malignancy Data Using Transformer-Based Models." Available at SSRN 5199243.*[9]
The paper applies transformer-based models to breast cancer clinical summaries for extracting key entities such as diagnoses, treatments, and biomarkers, enabling structured and accurate representation of medical information.

Workshop Papers

W1. Patra, Payel. *"Improving SoBigData Platform's FAIR Principles Through Decision-Tree and Web-Based Services."*. Presented at the 10th ACM Celebration of Women in Computing: womENcourage™ 2023, Trondheim, Norway, held at Trondheim Spektrum. The work proposes a decision-tree–based web service for diagnosing SoBigData's FAIR compliance and guiding users in addressing gaps to improve metadata quality, interoperability, and data reuse across the platform.

W2. Payel Patra, Daniele Di Pompeo, and Antinisca Di Marco, *"FAIR Tools: A Comparative Analysis and Their Role in Advancing Open Science,"* presented at the QualITA 2024 Workshop, Venice, June 13. presented at the Innovation Center Florence Workshop. Additional details available at:
`https://innovationcenterfirenze.it/en/`.

Publications not included in this thesis

C1. Patra, P., Bianchi, A., Di Pompeo, D., & Di Marco, A. (2024, March). Echocardiographic Epicardial Adipose Tissue Quantification: Challenges and Insights. *In 2024 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops) (pp. 619-624). IEEE. The paper focuses on automating the delineation of epicardial adipose tissue (EAT) in echocardiogram videos as a step toward more reliable and reproducible thickness quantification of EAT.*

[8] `https://arxiv.org/abs/2503.15929`
[9] `https://papers.ssrn.com/sol3/papers.cfm?abstract_id=5199243`

Journal Articles (Manuscripts in Preparation)

J1. Payel Patra, Antinisca Di Marco, and Nguyen, Phuong T., *Entity Extraction in Clinical Summary of Mammary Malignancy Data Using Transformer-Based Models.* Manuscript in preparation for submission to *International Journal of Data Science and Analytics*.

J2. Payel Patra, Daniele Di Pompeo, and Antinisca Di Marco. *A Comparative Study and Evaluation of Different FAIR Assessment Tools and their Implications in Open Science.* Manuscript in preparation for submission to *IEEE Transactions on Emerging Topics in Computational Intelligence (TETCI).*

J3. Payel Patra, Antinisca Di Marco, and Nguyen, Phuong T., *Integrating Medical Entity Knowledge Graphs with GNN-Based Pattern Analysis for Reinforcement Learning–Guided Medicine Optimization (RLGMO).* Manuscript in preparation for submission to *Artificial Intelligence in Medicine*.

1.3 Thesis outline

This thesis is organized into three main parts, each addressing a distinct dimension of the research.

- Part I of the thesis focuses on Medical Data Analysis, which transforms unstructured clinical data into structured and interpretable knowledge using artificial intelligence. This part introduces our MetaGRO Framework *(Medical Entity and Text Analysis for Graph-based Reinforcement Optimization)*, covering its main components: entity extraction, knowledge graph construction, and therapy recommendation techniques. Chapter 2 introduces the background of medical entity extraction, knowledge graph representation, and reinforcement learning–based decision systems. Chapter 3 reviews related work on AI in healthcare data management, particularly focusing on named entity recognition (NER), knowledge graphs (KG), and therapy recommendation systems. Chapter 4 presents the methodology and evaluation of the NER model applied in this research, while Chapter 5 explains the design and implementation of the medical knowledge graph and discusses the reinforcement learning–based therapy recommendation framework and analyzes its results.

- Part II focuses on the foundation of open science and the evaluation of FAIR assessment tools. It introduces the conceptual background and explores methodologies for ensuring data transparency and interoperability across research infrastructures. Chapter 6 presents the background concepts of open science, FAIR principles, and FAIR-enabling tools with the reviews of related studies on the FAIR-a tools' evaluation framework. Chapter 7 examines different open science platforms and proposes strategies to improve their FAIR compliance. Chapter 8 outlines the methodology for evaluating FAIR-a tools and provides detailed information on the evaluation procedures. It concludes with a discussion of the results and key insights derived from the FAIR tool assessment.

- Chapter 9 presents the discussion and interpretation of the results obtained in this study. It provides an in-depth analysis of the findings related to the AI-driven medical data analytics framework, the evaluation of FAIR assessment tools, and the echocardiographic EAT quantification. The chapter also offers an overall reflection on the study, including the system-level integration of FAIR

principles within the MetaGRO framework. Finally, Chapter 10 concludes the thesis by summarizing the key findings and contributions of the research, presenting the lessons learned, outlining potential directions for future work, and discussing the clinical deployment scope and ethical considerations of the proposed framework.

Part I

Medical Data Analysis (MDA)

Chapter 2

Background on MDA

This chapter introduces the background concepts that support the Medical Data Analysis (MDA) research conducted in this study. It first explains the concept of medical and clinical entity extraction in Section 2.1, highlighting how unstructured clinical texts are transformed into structured data for data analysis. The chapter then traces the evolution of these techniques—from early rule-based and statistical models to modern transformer-based architectures such as domain-specific BERT models and their biomedical extensions. In addition, this section provides an overview of the core BERT architecture, which underlies the domain-adapted models employed in this research, including BioBERT, BioClinicalBERT, and BlueBERT. Finally, it introduces the concepts of medical knowledge graphs (MKG) in Section 2.2 and medication recommendation systems in Section 2.3, showing how these elements together support the development of our proposed MetaGRO framework.

2.1 Medical and Clinical Entity Extraction

Medical data analysis focuses on extracting and interpreting meaningful information from diverse biomedical data sources to enhance both research and clinical decision-making. Within this domain, the closely related tasks of—**medical entity extraction** and **clinical entity extraction**—are essential for transforming unstructured biomedical or clinical text into structured, machine-readable representations.

Medical Entity Extraction (MEE) involves identifying and classifying biomedical concepts—such as diseases, symptoms, medications, and anatomical terms—from scientific and biomedical text sources. It is a fundamental biomedical natural language processing (BioNLP) task that converts unstructured text into structured, machine-readable information. *For example, extracting entities such as the disease "breast cancer," the drug "aspirin," or "chemotherapy" from PubMed abstracts or research articles.*

Clinical Entity Extraction (CEE) [26] is a specialized form of medical entity extraction focused on *clinical narratives* such as EHRs, discharge summaries, radiology reports, and clinician notes. It identifies patient-specific information—including diagnoses, symptoms, medications with their dosages, cancer stages, and treatment outcomes— supporting personalized medicine and clinical decision support. *For instance, detecting phrases like "Patient diagnosed with Stage II breast cancer" or "prescribed 50 mg of tamoxifen daily."* – doctor's note.

2.1.1 Evolution of Medical and Clinical Entity Extraction

Although this study focuses on **Clinical Entity Extraction (CEE)**, the task comes from the broader area of **Medical Entity Extraction (MEE)**. Many CEE methods and datasets were first developed for biomedical NER. Therefore, it is useful to look

at how medical entity extraction has developed before explaining the CEE method used in this study.

Medical entity extraction, or **clinical named entity recognition (NER)**, has advanced substantially over time. It aims to automatically identify biomedical concepts in unstructured text using NLP and AI techniques. This field began with the Message Understanding Conferences (MUC) in the 1990s [27], [28], which introduced the first approaches to information extraction and sequence labelling. In biomedical NLP, tools such as *MetaMap* and UMLS [29], [30] helped convert free-text expressions into standard terminology, improving consistency and interoperability. Annotated resources (corpora) like the *GENIA* corpus [31] additionally facilitated the advancement of supervised biomedical NLP models.

A major milestone in clinical NER was the 2010 *i2b2* challenge on extracting medical concepts from de-identified clinical notes [32]. It introduced standard entity categories (*problem*, *treatment*, *test*) and widely adopted evaluation metrics (precision, recall, and F1-score), establishing a reproducible benchmarking framework still influential today to shape clinical NLP research.

Initially these above-mentioned approaches were dominated by rule-based and dictionary-driven methods. Over time, these were supplemented by statistical sequence models such as Conditional Random Fields (CRFs) [33], Hidden Markov Models (HMMs), and Support Vector Machines (SVMs), which offered improved flexibility and generalization. While these statistical models provided interpretability, they depended heavily on manual feature engineering and faced scalability limitations. The emergence of deep learning architectures, particularly Bidirectional Long Short-Term Memory (BiLSTM) networks with CRF (Conditional Random Field) layers, represented a significant progression by enabling automatic modelling of contextual dependencies in clinical text.

2.1.2 Machine Learning overviews for Medical Data Analysis

Machine learning (ML) approaches are the base of the modern medical data analysis. In the beginning, traditional ML models were used vastly, such as **Support Vector Machines (SVMs)** and **Random Forests (RFs)** [34], [35]. These models depended on experts' decisions on manually selected features, where they had to decide which patterns or terms were important to consider. Although these models were simple and easy to interpret and understand, these models in reality could not fully capture the complex nature of medical language.

With the emergence of **deep learning**, architectures such as **Convolutional Neural Networks (CNNs)** and **Recurrent Neural Networks (RNNs)** [36], [37] enabled automatic feature learning directly from data, reducing reliance on manual feature design. This was followed by the adoption of **Long Short-Term Memory (LSTM)** networks [38] and their bidirectional extension (**BiLSTM**), which became widely used in biomedical text processing due to their ability to retain and utilize long-range contextual information.

A major improvement came with the rise of **Transformer architectures** [39]. In contrast to other sequence-based models such as RNNs and LSTMs, transformer models utilize a self-attention mechanism to evaluate the relationships among all tokens in parallel. This characteristic makes them highly effective for interpreting long and information-dense clinical documents.

Transformers usually work in two stages:

1. **Pre-training:** The model is first trained on very large biomedical datasets such as PubMed or MIMIC-III [40], [41] to learn the general meaning and structure of biomedical language.

2. **Fine-tuning:** After pre-training, the model is trained again (fine-tuned) on smaller, task-specific datasets—like identifying diseases, medications, or symptoms—to perform clinical text extraction accurately for specific purposes.

This two-step learning process is the main reason for the success of models like **BERT** [42] and its biomedical versions, including **BioBERT** [41], **BioClinicalBERT** [43], **BlueBERT** [44], and **PubMedBERT** [45]. These models can handle the complex vocabulary found in clinical documentation and learn deep semantic connections among the medical terms by first learning general language patterns and then adjusting to biomedical corpora. As a result, transformer-based methods are now essential for converting unstructured clinical language into organized, useful data for healthcare analytics.

2.1.3 Advances in Transformer-Based Models for Entity Extraction

Transformer-based models have played a major role in the recent evolution of medical and clinical entity extraction. Earlier approaches, like explained before, including rule-based systems and traditional machine learning techniques, showed difficulties in capturing long-range contextual information in clinical narratives. In contrast, modern transformer architectures such as **BERT** and its biomedical variants—**BioBERT, BioClinicalBERT, BlueBERT, and PubMedBERT**—have substantially advanced the field. Using self-attention and pre-training on corpora like *PubMed* and *MIMIC-III*, these models learn rich contextual patterns and can detect complex clinical entities with improved accuracy, even in specialized or varied language. Fine-tuned on clinical annotations, transformer models achieve state-of-the-art performance in extracting diseases, symptoms, medications, and treatment details. Moving from older methods to these context-aware models is a major step forward for clinical NLP. Their structured outputs also support downstream healthcare analysis.

Bidirectional Encoder Representations from Transformers (BERT)

The **BERT** model [42] marked a major breakthrough in natural language processing. Unlike earlier one-directional models, it uses a bidirectional Transformer encoder that reads context from both sides of a sentence, enabling it to capture complex relationships and subtle meanings [39]. BERT is trained using two key objectives:

- **Masked Language Modeling (MLM):** Randomly hides certain words in a sentence and trains the model to predict them based on surrounding context.

- **Next Sentence Prediction (NSP):** Learns whether one sentence logically follows another, improving sentence-level understanding.

Through these objectives, BERT learns strong and general language representations that can later be fine-tuned for domain-specific tasks such as medical or clinical entity extraction. Several domain-adapted versions have since been introduced, including *BioBERT* [41], *BioClinicalBERT* [43], and *PubMedBERT* [45], which use the same BERT architecture but are trained on biomedical and clinical corpora. These

models have significantly improved the accuracy and contextual understanding of medical language processing tasks.

The transition from rule-based and statistical methods to transformer-based architectures has established a strong foundation for modern medical data analysis. The next section explains how structured information extracted from clinical text contributes to medical knowledge graph construction and medication recommendation frameworks.

BERT (Bidirectional Encoder Representations from Transformers) [42] is derived from the Transformer framework proposed by Vaswani et al. [39]. Only the encoder presented in Figure 2.1, is used by the BERT model, which uses a bidirectional self-attention mechanism to model contextual relationships within text data. The original Transformer architecture consists of an encoder–decoder structure stated below. The BERT model achieves state-of-the-art performance in classification, entity recognition, and another set of question-answering tasks when trained with the Masked Language Modeling (MLM) objective. On the other hand, architectures like T5 (Text-To-Text Transfer Transformer) [46] and BART (Bidirectional and Auto-Regressive Transformer) [47] maintain the complete encoder–decoder design, making them more appropriate for generative NLP tasks. The encoder layer and decoder layer formulation follows the original Transformer design introduced by Vaswani et al. [39].

Transformer Encoder: It is a stack of transformer layers that performs bidirectional input text processing. A feedforward network is used to improve these representations after self-attention is used to record the associations between each character in the sequence. To get a thorough comprehension of the text, the encoder generates contextual embeddings for every token.

Encoder Formula

$$H^{(l+1)} = \text{LayerNorm}\left(H^{(l)} + \text{MultiHeadAttention}(H^{(l)})\right) + \text{LayerNorm}\left(\text{FFN}(H^{(l)})\right)$$

Where: $H^{(l)}$ represents the input to the l-th encoder layer. The **MultiHeadAttention** mechanism captures relationships between tokens using the scaled dot-product attention mechanism. **FFN** (Feedforward Network) is applied independently to each token for further processing. Finally, **LayerNorm** (Layer Normalization) stabilizes the output and improves training efficiency. All the models are pure encoder models and do not have a decoder component, being optimized for feature extraction, not for text generation.

Transformer Decoder: Using both the encoder's output (by encoder-decoder attention) and the previously created tokens (via masked self-attention), the decoder, which is employed in models such as GPT, produces output sequences. This process enables it to produce text sequences that are both logical and contextually aware. However, BERT does not come with a decoder; it entirely concentrates on encoding.

Decoder Formula

$$
\begin{aligned}
H^{(l+1)} = & \text{LayerNorm}\left(H^{(l)} + \text{MaskedMultiHeadAttention}(H^{(l)})\right) \\
& + \text{LayerNorm}\left(H^{(l)} + \text{MultiHeadAttention}(E, H^{(l)})\right) \\
& + \text{LayerNorm}\left(\text{FFN}(H^{(l)})\right)
\end{aligned}
$$

Where: $H^{(l)}$ represents the input to the l-th decoder layer, while E is the output of the encoder. The **MaskedMultiHeadAttention** mechanism ensures causal masking, preventing future tokens from being attended to. The **MultiHeadAttention** mechanism applies attention over the encoder output E and the decoder input $H^{(l)}$. The **FFN** (Feedforward Network) processes the decoder output independently for each token. Finally, **LayerNorm** (Layer Normalization) is applied after each sublayer to stabilize training and improve performance. Each encoder layer consists of a *multi-head self-attention mechanism*, *layer normalization*, and a *feed-forward network*, connected through residual links. This architecture enables BERT to attend to every token in a sequence in both forward and backward directions simultaneously, resulting in richer contextual understanding for natural language processing tasks.

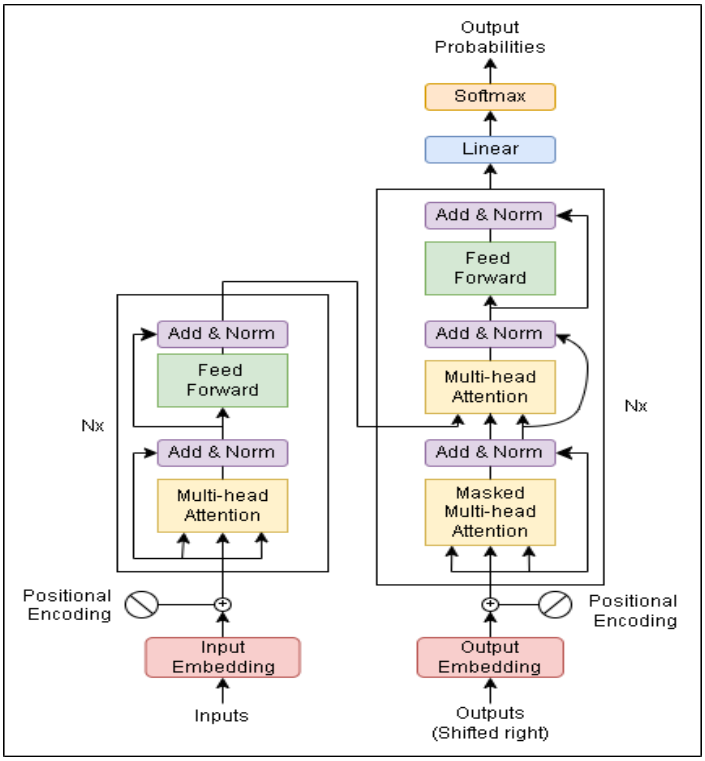


FIGURE 2.1: The Transformer model architecture consists of an encoder (left) and decoder (right) stack [39]. BERT utilizes only the encoder component.

Before looking at the biomedical versions of BERT, it helps to see how the original BERT model is built. The main parts of its architecture, as shown in Figure 2.1, are explained in the following paragraphs.

Input Representation: BERT represents each input token using a combination of three embeddings: *token embeddings* that capture the meaning of each word or subword, *segment embeddings* that distinguish between paired sentences, and *positional embeddings* that encode the order of tokens in the sequence. These three components are summed to form the final input representation.

Self-Attention Mechanism: BERT's core component is the self-attention mechanism, allowing each token to attend to all others in the sequence. By comparing query, key, and value vectors, the model computes attention weights and captures both short- and long-range context.

Layer Normalization: BERT improves training stability by applying layer normalization after each encoder sublayer. By normalizing hidden states using their mean and variance and applying learnable scaling and shifting, the model becomes less sensitive to changes in its internal activations.

Feed-Forward Network: Within each encoder layer, a position-wise feed-forward network applies two linear transformations with a non-linear activation in between. Operating independently on each token, this network increases the model's expressive capacity and enables it to capture more complex patterns than self-attention alone can provide.

Activation Function: BERT uses the Gaussian Error Linear Unit (GELU) as its activation function, offering a smoother and more probabilistic alternative to ReLU. GELU improves the model's capacity to capture subtle language variances by weighting inputs according to both their value and related likelihood of occuring.

Model Outputs: Given an input sequence of tokens, BERT produces context-rich output embeddings in which each token representation incorporates information from the entire sentence. These contextualized embeddings are then used in downstream tasks such as text classification, named entity recognition (NER), and question answering.

Summary: BERT's architecture consists of 12 encoder layers in $\text{BERT}_{\text{Base}}$ and 24 encoder layers in $\text{BERT}_{\text{Large}}$. Each layer models deep bidirectional dependencies across tokens, forming a strong foundation for domain-specific variants such as BioBERT, ClinicalBERT, PubMedBERT, BlueBERT, and RoBERTa. While these variants maintain the original architectural design, they differ in their pre-training datasets, enabling improved performance on biomedical and clinical text understanding tasks.

BERT-based model Selection

A brief description of all the selected BERT-based models used in our thesis work is provided below, and their key differences are summarized in Table 2.1.

BioBERT: BioBERT is a domain-specific extension of the original BERT architecture [42], shown in Figure 2.2, developed to better handle the terminology and linguistic complexity of biomedical text. The model is first initialized with general-domain BERT weights and then further pre-trained on large biomedical corpora,

including PubMed abstracts and PMC full-text articles [41]. This extended pre-training—performed using the same Masked Language Modeling (MLM) and Next Sentence Prediction (NSP) objectives at the same time—enables BioBERT to capture domain-specific expressions more effectively than general-purpose models. Like BERT, it uses a WordPiece tokenizer, which helps reduce the Out-of-Vocabulary (OOV) problem by splitting rare biomedical terms into meaningful subword units (e.g., *Immunoglobulin* → "I ##mm ##uno ##g ##lo ##bul ##in"). Owing to this domain adaptation, BioBERT achieves higher accuracy in biomedical entity recognition, relation extraction, and question answering compared to other BERT variants commonly used in medical NLP, as summarized in Table 3.2.

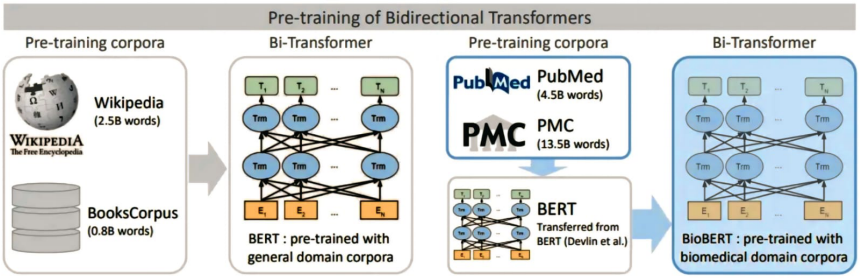


FIGURE 2.2: Overview of the BioBERT model.[1]

BioClinicalBERT: BioClinicalBERT [43] expands upon BioBERT by extending clinical-domain knowledge through extra pre-training on the MIMIC-III electronic health record (EHR) corpus. It can capture the abbreviations, real-world language, and shorthand idioms frequently found in clinical texts, whereas BioBERT is largely focused on biomedical literature. It keeps the original BERT architecture and training goals (MLM and NSP), but because it has access to clinical notes, it can identify patterns related to the diagnosis, symptoms, drugs, and treatment specifics that medical professionals have recorded. This makes BioClinicalBERT particularly effective for tasks such as clinical entity extraction, clinical relation categorization, and other applications requiring EHR text.

PubMedBERT: PubMedBERT [45] is quite different from existing biological BERT variants since it is trained from scratch on PubMed abstracts and full-text biomedical publications, utilizing a vocabulary derived entirely from biomedical corpora. In contrast to BioBERT, which builds upon general BERT weights, PubMedBERT applies a fully domain-exclusive training technique that boosts its potential to capture biomedical terminology, morphological variants, and specialized technical phrases. The model is trained only with the MLM objectives, representing studies that undermine the relevance of the NSP task. PubMedBERT achieves better performance in tasks like biomedical NER, relation extraction, and document classification as a result of its domain-focused methodology.

BlueBERT: BlueBERT [44] is a biomedical adaptation of BERT trained on PubMed abstracts and MIMIC-III clinical notes. These two sources of training allow the model to bridge differences between biomedical research discourse and clinical language usage. Like these models, BERT and BioBERT, BlueBERT uses the MLM and

[1] `https://medium.com/@raghudeep/biobert-insights-b4c66fde8fa7`

NSP objectives and incorporates WordPiece tokenization to effectively manage specialized medical vocabulary. As a result, it performs strongly on a variety of biomedical and clinical NLP tasks, particularly those that require understanding across both domains.

RoBERTa: RoBERTa [48] is an enhanced BERT variant that eliminates the NSP objective, applies dynamic masking, and benefits from training on significantly larger datasets with longer optimization schedules. While RoBERTa itself is not tailored to biomedical text, domain-adapted extensions such as BioRoBERTa apply additional pre-training on biomedical corpora such as PubMed and MIMIC-III. These adaptations combine RoBERTa's improved training strategy with biomedical domain knowledge, allowing them to deliver strong results across tasks including clinical NER, text classification, and relation extraction. It is basically pretrained on CC-News, OpenWebText, and BooksCorpus, removes NSP, and uses dynamic masking per epoch.

TABLE 2.1: Pre-training details for selected BERT-based models, including corpus, word count, and domain.

Model	Pre-Training Corpus	Number of Words	Domain
BERT (Base model)	Wiki + Books	3.3 billion	General
BioBERT (+ PubMed + PMC)	Wiki + Books + PubMed + PMC	20.3 billion	Biomedical
BioClinicalBERT	MIMIC-III + Clinical Notes (EMR) + PubMed + PMC	N/A	Biomedical/Clinical
PubMedBERT	PubMed abstracts + full texts	16 billion	Biomedical
BlueBERT	PubMed abstracts + MIMIC-III Clinical Notes	N/A	Biomedical/Clinical
RoBERTa (Baseline)	Common Crawl + Books + OpenWebText + Stories	160GB of text	General

2.2 Medical Knowledge Graphs

A **knowledge graph (KG)** is a structured data model that represents entities (such as people, places, or objects) and the relationships among them in a graph form [49]. It allows information to be connected through semantic relationships, enabling both humans and machines to understand and reason over data. Knowledge graphs are widely used in artificial intelligence and data management for applications like search engines, recommendation systems, and intelligent assistants, as they provide an interpretable structure for storing and querying complex relationships.

Building on this general concept, a **Medical Knowledge Graph (MKG)** represents the relationships among medical entities and their attributes [50], [51]. By combining data from a variety of sources, including structured and unstructured data, electronic health records, published biomedical literature, and ontologies, MKGs provide an organized representation of knowledge. Graph convolutional neural networks are used for the following purposes: Clinical decision support systems, drug repurposing, and individualized therapy recommendations were among the medical fields in which they were used [52], [53], [54], [55], [56], [57], [58].

As presented in Figure 2.3, a medical knowledge graph relates the identified entities, including diseases, symptoms, lab tests, and medications, to create a meaningful semantic graph. In the diagram, it is clear that the transformer-based model has a bidirectional interaction with the knowledge graph. This is necessary for information extraction and reasoning on clinical data.

A **Healthcare Knowledge Graph (HKG)** builds on this concept by highlighting the importance of patient-centric relations, connecting diseases, medications, and

symptoms in a manner that offers a clearer, more informed view of clinical information [59], [60]. HKGs enable the linking of diverse data sources to build a common and interpretable platform for healthcare analytics.

In this research, the proposed **Reinforcement Learning–Guided Medicine Optimization (RLGMO)** module serves as a key intelligent component within the MetaGRO Framework and is grounded in the concept of Healthcare Knowledge Graphs (HKGs). Unlike general medical knowledge graphs that capture broad biomedical relationships, the HKG constructed in this study is centered on patient-specific data and models the clinical interactions among diseases, symptoms, and medications. This structured representation enables RLGMO to reason over individual patient profiles and optimize personalized medication recommendations, thereby contributing to effective clinical decision support.

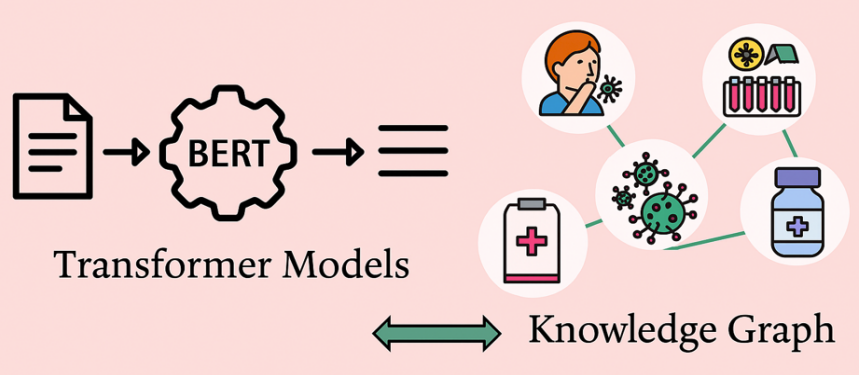


FIGURE 2.3: An overview of the interaction between unstructured data, transformer models, and medical knowledge graphs. The model extracts and links entities such as diseases, drugs, and laboratory findings.

2.3 Medication or Therapy Recommendation

Medication or treatment recommendation approaches generally fall into three categories: ***(i)*** rule-based and guideline-driven systems, ***(ii)*** supervised learning methods that predict outcomes for candidate therapies, and ***(iii)*** reinforcement learning (RL) approaches that learn treatment policies from longitudinal patient data [61], [62], [63]. Recent research increasingly represents patients, diseases, and treatments as graphs, allowing graph neural networks (GNNs) to capture relationships and dependencies between medical variables [64], [65]. These methods closely relate to Clinical Decision Support Systems (CDSS), which provide the socio-technical layer — including user interfaces, audit mechanisms, and governance—through which clinical recommendations are delivered [66], [67].

From Rules to Learning Rule-based and guideline-driven systems provide consistency but often fail to handle the complexity and variability of real-world treatment scenarios [68]. Supervised learning methods can predict treatment outcomes but are limited to static datasets and do not optimize sequential decisions. Reinforcement learning, on the other hand, focuses on learning through interaction—adapting treatment strategies based on feedback from patient outcomes [69], [70]. When combined with GNNs, RL-based systems can model the complex relationships among diseases, symptoms, and medications to make personalized and dynamic recommendations [71].

Position of This Thesis. Based on the above discussion, this thesis adopts a reinforcement learning (RL) perspective for treatment optimization. The proposed **Reinforcement Learning–Guided Medicine Optimization (RLGMO)** module acts as the central intelligent component of the MetaGRO Framework, combining healthcare knowledge graphs with graph neural networks to learn medication policies from structured clinical data. Within MetaGRO, RLGMO handles adaptive policy learning and personalized treatment optimization, while the remaining components perform entity extraction, knowledge organization, and FAIR-aligned data management.

Although MetaGRO is not yet a complete *Clinical Decision Support System* (CDSS), its modular design provides a strong foundation for future extensions toward a transparent, interpretable, and data-driven CDSS. The concepts outlined in subsection 2.3.1 directly motivate the design of the RLGMO module. Here in this section, the architecture is discussed in detail, including how the Graph Neural Network (GNN), realized through a Graph Convolutional Network (GCN), works together with the Deep Q-Network (DQN) to produce personalized therapy recommendations. This subsection also outlines the background idea of reinforcement learning setting used in this work, including the definitions of agents, states, actions, and rewards, and highlights Deep Q-Learning as the selected policy optimization method for graph-based treatment strategy learning.

2.3.1 Foundations of GCN–DQN Architectures

The development of data-driven models for clinical decision assistance has improved dramatically, thanks to artificial intelligence. Traditional deep learning models efficiently extract intricate patterns from huge datasets, but they frequently fall short in two crucial areas for medical decision-making: *explainability* and *adaptability*. In order to address these issues, we develop RLGMO (Reinforcement Learning Guided Medicine Optimization), an approach that combines structured medical information, graph-based learning, and reinforcement learning (RL) to offer therapeutic suggestions that are both individualized and comprehensible.

In our MetaGRO framework, the final component of the RLGMO model integrates knowledge graph construction using extracted entities from Cancer Clinical Text Data, GCN-based pattern extraction and training, and reinforcement learning to generate therapy recommendations. After that, cosine similarity is used to validate semantic similarity between different medications of the same therapeutic class, ensuring clinically meaningful outputs.

Reinforcement Learning (RL) Taxonomy

Reinforcement learning algorithms are broadly categorized into model-based and model-free approaches. As discussed in Figure 2.4, model-free algorithms do not require an explicit model of the environment. RLGMO adopts a model-free framework, specifically the Q-learning family, due to its simplicity and effectiveness in discrete action spaces. We use DQN for its stability and scalability when combined with neural networks. This makes it ideal for optimizing therapy recommendations using GCN-derived patient representations.

[4]Source: `https://spinningup.openai.com/en/latest/spinningup/rl_intro2.html`

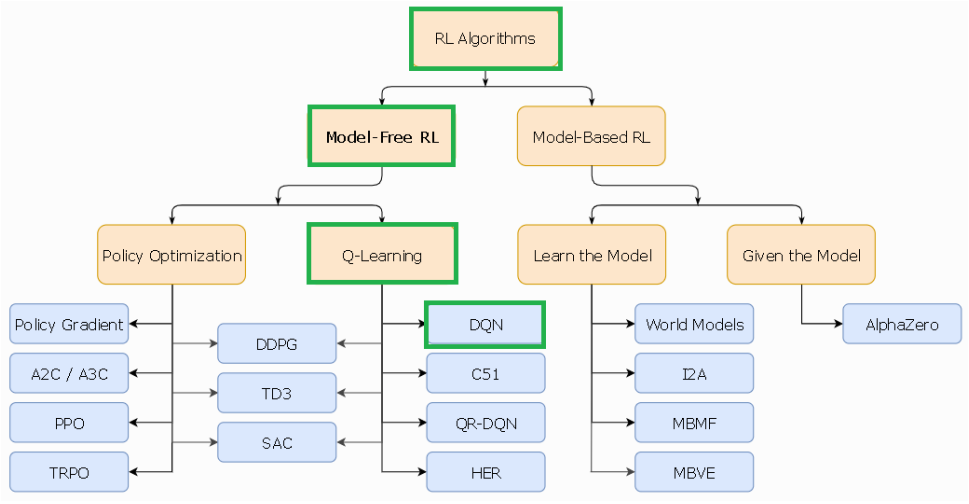


FIGURE 2.4: A non-exhaustive taxonomy[4] of reinforcement learning algorithms. The highlighted path represents the direction taken in RLGMO.

Preliminaries

Given a patient p_i, we construct a personalized knowledge graph $G_i = (V_i, E_i)$, where V_i is the set of clinical entities—including diseases, symptoms, medications, dosages, and medical history—and E_i represents the semantic relationships among them. These graphs encode structured views of each patient's clinical profile, forming the foundation for graph-based representation learning.

To embed these graphs for downstream reasoning and therapy optimization, we employ **Graph Convolutional Networks (GCNs)** [72]. GCNs learn node representations by iteratively aggregating features from a node's neighbors. At each layer l, the embedding of a node v is updated using :

$$h_v^{(l+1)} = \sigma\left(\sum_{u \in \mathcal{N}(v)} \frac{1}{c_{vu}} W^{(l)} h_u^{(l)}\right) \tag{2.1}$$

where $h_v^{(l+1)}$ is the updated embedding of node v, $W^{(l)}$ is the trainable weight matrix at layer l, c_{vu} is a normalization constant (e.g., based on node degrees), and σ is a non-linear activation function such as ReLU.

The matrix form of the update is :

$$H^{(l+1)} = \sigma\left(\tilde{D}^{-1/2}\tilde{A}\tilde{D}^{-1/2}H^{(l)}W^{(l)}\right) \tag{2.2}$$

where $\tilde{A} = A + I$ is the adjacency matrix with self-loops, $\tilde{D}$ the corresponding degree matrix, $H^{(l)}$ the node feature matrix, and $W^{(l)}$ the trainable weight matrix. The initial features $H^{(0)}$ can be either one-hot encodings or domain-specific embeddings of medical concepts.

Therapy Optimization as a Markov Decision Process

We frame the therapy recommendation task as a **Markov Decision Process (MDP)** [73]:

$$\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \gamma) \tag{2.3}$$

→ $\mathcal{S}$: the state space, represented by GCN-based patient embeddings;

→ $\mathcal{A}$: the action space, consisting of therapy or medication options;

→ $\mathcal{T}$: the transition function (unknown and implicitly modeled);

→ $\mathcal{R}$: the reward function, reflecting treatment relevance;

→ γ: the discount factor for future rewards.

The agent learns a policy $\pi(\theta)$, parameterized by θ, to maximize the expected cumulative reward:

$$\pi^*(\theta) = \arg\max_{\pi} \mathbb{E}_{\pi} \left[\sum_{t=0}^{\infty} \gamma^t r_t \right] \tag{2.4}$$

Here:

→ $\pi(\theta)$ is the policy that maps states to actions;

→ $\mathbb{E}_{\pi}$ is the expectation over trajectories induced by the policy;

→ r_t is the reward at time step t;

→ $\gamma \in [0, 1)$ controls the trade-off between immediate and future rewards.

In our proposed **RLGMO (Reinforcement Learning Guided Medicine Optimization)**, each state s_t corresponds to a graph-based embedding of the patient's profile, and each action a_t refers to a recommended treatment. The agent is trained to identify interventions that maximize reward across various patient scenarios.

The reward signal r_t is calculated based on the *cosine similarity* between the recommended therapy and known effective treatments in similar cases. To ensure medically valid comparisons, we use the *RxNorm*[5] drug classification system, which enables identification of therapeutically equivalent drugs even when brand names or formulations differ.

Deep Q-Learning for Policy Optimization

In our RLGMO stage, we employ **Deep Q-Learning (DQN)** [74]—a model-free, value-based reinforcement learning algorithm—to train an agent for therapy recommendation. DQN is particularly effective when the action space is discrete, such as selecting a medication or treatment from a finite set.

The goal of DQN is to learn a function $Q(s, a; \theta)$ that estimates the expected cumulative reward of taking action a in state s, and then following the learned policy thereafter. Here, θ represents the parameters of a neural network (the Q-network) used to approximate the Q-function.

The expected return is computed using the Bellman equation:

$$Q(s, a; \theta) = \mathbb{E}\left[r + \gamma \max_{a'} Q(s', a'; \theta^-) \mid s, a \right] \tag{2.5}$$

where:

→ r is the immediate reward received after taking action a,

[5] RxNorm documentation: `https://www.nlm.nih.gov/research/umls/rxnorm/`

→ $\gamma \in [0, 1)$ is the discount factor for future rewards,

→ s' is the next state resulting from action a,

→ θ^- denotes the parameters of a separate target network, which is periodically updated to stabilize training,

→ $\max_{a'} Q(s', a'; \theta^-)$ represents the maximum predicted reward achievable from the next state.

To train the Q-network, we minimize the **temporal difference (TD) loss**:

$$L(\theta) = \mathbb{E}_{(s,a,r,s')} \left[\left(r + \gamma \max_{a'} Q(s', a'; \theta^-) - Q(s, a; \theta) \right)^2 \right] \tag{2.6}$$

This loss measures the difference between the current Q-value and a better estimate based on the reward and next state. The network parameters are updated via gradient descent:

$$\theta \leftarrow \theta + \alpha \left(r + \gamma \max_{a'} Q(s', a'; \theta^-) - Q(s, a; \theta) \right) \nabla_\theta Q(s, a; \theta) \tag{2.7}$$

where α is the learning rate, and $\nabla_\theta Q(s, a; \theta)$ is the gradient of the Q-function with respect to the network parameters.

By continuously observing transitions (s, a, r, s') and refining the Q-function, the agent learns to recommend therapies that maximize long-term outcomes. This process enables intelligent treatment planning based on the patient's graph-based clinical state.

In the context of this thesis, the state s_t represents the graph-based embedding of a patient constructed from extracted clinical entities and their relationships within the knowledge graph. The action a_t corresponds to a potential therapy or medication recommendation selected by the RLGMO agent. The reward signal reflects the similarity between the recommended therapy and validated historical treatments or reference guidelines, thereby encouraging clinically coherent and evidence-aligned decisions. This formulation enables reinforcement learning to operate over structured patient representations derived from real-world oncology data.

Chapter 3

Related Work

This chapter reviews relevant studies across four major areas. Section 3.1 presents existing work on artificial intelligence in healthcare data management, emphasizing methods that enhance the organization, integration, and analysis of complex medical data. Section 3.2 summarizes approaches for named entity recognition (NER) in clinical text, highlighting strategies for extracting structured medical knowledge from unstructured sources. Section 3.3 describes the research on knowledge graphs in biomedical informatics, focusing on their ability to model intricate relationships among medical entities and support reasoning tasks. Finally, Section 3.4 reviews studies employing reinforcement learning for therapy recommendation, with attention to adaptive, personalized, and data-driven clinical decision-making.

3.1 AI in Healthcare Data Management

Artificial Intelligence (AI) has been growing to become a fundamental pillar in the current healthcare environment, facilitating the processing and analysis of large datasets in the fields of healthcare, biomedical subjects, and administrative matters for effective patient management and healthcare delivery [75, 76]. In the management of healthcare data for effective patient monitoring and healthcare delivery, AI models that incorporate simple machine learning algorithms to sophisticated deep learning systems have been utilized to make healthcare-related decisions and provide patient monitoring and treatment suggestions for effective patient healthcare delivery [77, 78]. In hospital information systems, AI applications assist in dealing with electronic health records, assist in interoperability between different platforms, and improve search features for accessing specific health-related data [79, 80]. Such applications lead to a reduction in errors, improve accessibility of health-related data, and assist in rapid and precise identification of health-related diagnoses. Driven by AI, applications also assist in handling aspects of population health management and trend analysis of health-related data [81].

In view of the significance of structured and unstructured data formulation and interpretation in the healthcare industry, more advanced natural language processing methods like Named Entity Recognition have become indispensable in extracting relevant health-related information from medical texts [82]. Extracted health-related information serves as a basis for constructing a knowledge graph as well as linking patient health records to a structured vocabulary to facilitate AI-based health-related decision support.

3.2 Named Entity Recognition in Clinical Text

Named Entity Recognition (NER) converts free-text clinical narratives into structured data by locating and classifying mentions of diseases, symptoms, medications, dosages, procedures, demographic attributes, and cancer-specific concepts such as stage and histopathology. Compared with general-domain NER, biomedical NER faces domain-specific terminology, heavy use of abbreviations, and frequent synonymy, which demand context-sensitive modeling [83, 84, 85].

Early clinical NLP toolchains (e.g., rule-based or dictionary-driven) remain valuable for concept mapping and annotation workflows, while modern transformer models provide state-of-the-art accuracy by learning contextual representations. Table 3.1 summarizes widely used tools for clinical text processing and annotation that support corpus creation and curation for NER model development.

TABLE 3.1: Comparison of Clinical Text Processing and Medical Entity Extraction Tools.

Tool Name	Available Links	Functionality	Exclusion Criteria from Others
BRAT	`https://brat.nlplab.org/`	Web-based text annotation and validation	Requires external setup, lacks built-in ML support
INcepTION	`https://inception-project.github.io/`	Collaborative annotation with machine learning assistance	Complex for beginners, requires ML expertise
MetaMap	`https://metamap.nlm.nih.gov/`	Maps biomedical text to UMLS concepts	Limited to UMLS concepts, lacks broader NLP capabilities
MedLEE	`https://www.dbmi.columbia.edu/research-projects/medlee/`	Structuring clinical narratives	Designed for clinical narratives, less flexible for general medical text
CLAMP	`https://clamp.uth.edu/`	Named entity recognition and relation extraction	Focuses mainly on named entity recognition, lacks deep inference
MedNLI	`https://physionet.org/content/mednli/1.0.0/`	Medical language inference and analysis	Limited to medical natural language inference tasks
BioMedGPT	`https://github.com/taokz/BiomedGPT`	Deep medical language processing	Primarily designed for deep learning-based medical text generation
MedspaCy	`https://github.com/medspacy/medspacy`	Extends spaCy for clinical text processing	Requires spaCy knowledge, limited pre-built clinical models
Doccano	`https://github.com/doccano/doccano`	User-friendly text annotation with JSON export support	Selected for its ease of use, efficient annotation workflow, and JSON export support (We selected)

Transformer-based NER. Transformer encoders (BERT and domain-specific variants) have driven rapid gains in biomedical NER by leveraging self-attention and large-scale pre-training. *BioBERT*, *ClinicalBERT* / *BioClinicalBERT*, *BlueBERT*, and *PubMedBERT* consistently outperform general BERT on biomedical corpora due to domain-matched pretraining [41, 42, 44, 45, 86]. Table 3.2 provides an overview of commonly used models and their pretraining sources.

TABLE 3.2: Overview of BERT Models, Pretraining Datasets, and Selection Criteria.

Model Name	Description	Pretraining Dataset(s)	Included	Criteria
BERT [87]	Original BERT model by Google for general NLP tasks.	BookCorpus + English Wikipedia	No	Not optimized for biomedical or legal domain-specific tasks.
BioBERT [88]	Specialized for biomedical text.	PubMed abstracts + PMC full-text articles	Yes	Designed specifically for biomedical applications.
ClinicalBERT [89]	Tailored for clinical data processing.	MIMIC-III (clinical records)	No	Limited to clinical records; lacks broader biomedical scope.
SciBERT [90]	Designed for scientific and research text.	Semantic Scholar corpus	No	Not fine-tuned for clinical or biomedical applications.
BlueBERT [91]	Combines biomedical and clinical data.	PubMed abstracts + MIMIC-III	Yes	Optimized for biomedical domain using both PubMed and MIMIC-III.
RoBERTa [48]	Enhanced version of BERT.	BookCorpus + Wikipedia + CC-News + OpenWebText	Yes	Strong general NLP performance; suitable for medical fine-tuning.
DistilBERT [92]	Lightweight, faster BERT.	Same as BERT	No	Reduced model size leads to lower accuracy in domain-specific tasks.
ALBERT [93]	Parameter-efficient BERT.	BookCorpus + English Wikipedia	No	Not pretrained on medical or clinical domain data.
BioClinicalBERT [94]	Biomedical/clinical hybrid model.	MIMIC-III + PubMed abstracts	Yes	Strong domain match: clinical + biomedical corpora.
COVID-Twitter-BERT [95]	COVID-19 social media text.	COVID-19 tweets	No	Domain restricted to COVID-19 and social media; poor general biomedical coverage.
FinBERT [89]	Financial-domain text analysis.	Financial reports, news	No	Designed for financial text; unsuitable for biomedical tasks.
LegalBERT [96]	Legal domain text.	Legal documents/cases	No	Domain mismatch; trained for legal language only.
CamemBERT [97]	French language BERT.	OSCAR French corpus	No	French-only; not usable for English biomedical NLP.
M-BERT [98]	Multilingual (104 languages).	Wikipedia multilingual	No	Not optimized for biomedical domain tasks.
XLM-RoBERTa [99]	Cross-lingual transformer.	Common Crawl (100+ languages)	No	No biomedical specialization.
PubMedBERT [100]	Biomedical literature model.	Full PubMed abstracts	Yes	Very strong biomedical focus; excellent for literature-based tasks.

Benchmarks and shared tasks. Robust progress in clinical NER has been catalyzed by shared tasks and curated corpora that enable reproducible evaluation: i2b2/VA 2010 for clinical concept extraction [101]; BioCreative chemical–disease resources [102, 103]; ShARe/CLEF eHealth concept normalization [104]; and more recent efforts addressing nested entities and realistic clinical complexity [105]. Complementary initiatives (e.g., TREC Precision Medicine, CLEF eRisk, and the NCBI Disease Corpus) broaden the evaluation landscape across tasks and data sources [106, 107, 108, 109]. Multilingual benchmarks such as DrBenchmark emphasize rigorous, task-specific evaluation design and motivate future extensions beyond English [110].

Search strategy snippet (as used for breast cancer NER). Listing 3.1 represents the search string executed on the Scopus database[1], utilizing three sets of keywords: (i) **Cancer (Breast)** — "Breast Cancer" OR "Clinical Text Data" OR "Breast Cancer Data" OR "Cancer Text Data"; (ii) **Entity Extraction** — "*Entity Extraction" OR "Extracting"; and (iii) **NLP/LLM/BERT** — "Natural Language Processing" OR "NLP" OR "Large Language Model" OR "LLM*" OR "Artificial Intelligence" OR "AI" OR "BERT" OR "*BERT" OR "Transformer Model*".

LISTING 3.1: The Scopus advanced string search

```
TITLE-ABS (("Breast Cancer" OR "Clinical Text Data" OR "
   ↪ Breast Cancer Data" OR "Cancer Text Data")
AND ("*Entity Extraction" OR "Extracting")
AND ("Natural Language Processing" OR "NLP" OR "Large
   ↪ Language Model" OR "LLM*" OR "Artificial Intelligence
   ↪ "
OR "AI" OR "BERT" OR "*BERT" OR "Transformer Model*"))
AND PUBYEAR > 2020 AND PUBYEAR < 2025
```

TABLE 3.3: The total number of papers related to the topics.

	Cancer (Breast)	**Entity Extraction**	**NLP/LLM/BERT**
Cancer (Breast)	48,548		
Entity Extraction	282	46,757	
NLP/LLM/BERT	1,737	1,073	45,884

Table 3.3 reports the number of papers containing these keywords within the corresponding categories. The count of papers containing both keywords **(Entity Extraction and Cancer (Breast))** is 282, while the combination of **(NLP/LLM/BERT) and Cancer (Breast)** keywords returns 1,737 articles. Our primary target is to get the count of papers that included at least one keyword that matches from each set of **cancer (breast)**, **entity extraction**, and **any NLP/LLM/BERT method**. Although NLP, LLM, and BERT represent broad methodological families, they are grouped together here because our objective is to capture all relevant computational approaches used for entity extraction in cancer-related clinical text data; therefore, we consider any technique falling under these categories as part of the wider landscape that must be reviewed.

The initial selection process identified 39 papers that met the criteria, as shown in Table C.1, from which 35 were chosen based on their relevance to our work. This table presents state-of-the-art methods for extracting various medical entities (Cancer-Related Diseases, Symptoms, Medication, Stages/Classification, Medical History, Cancer Type/Features (density/size), Cancer Detection, Knowledge Extraction, Gene Related) using different approaches.

For cancer data, including breast cancer, using only BERT-based models for entity extraction, we found some relevant papers that are relevant to our approach, which is discussed below.

Research in the last few years has concentrated on improving the extraction and analysis of data (medical entity) related to breast cancer through the use of *AI* and *NLP* approaches. However, we have considered BERT models for evaluation work

[1] https://www.scopus.com/

in our paper. Zhou *et al.* [111] introduced *CancerBERT*, a fine-tuned BERT model with the customized vocabulary, achieving superior performance in extracting breast cancer phenotypes from clinical texts. The result shows that CancerBERT outperforms baseline models such as BERT, BioBERT, and BlueBERT in Named Entity Recognition (NER), with the manually curated version of the vocabulary achieving the highest F1 scores. This study highlights the importance of domain-specific pre-training in medical NLP and suggests that CancerBERT can improve clinical decision-making by providing more accurate and structured extractions of critical cancer-related information from unstructured text.

Solarte-Pabón *et al.* (2023) [112] developed a transformer-based Named Entity Recognition (NER) system to extract breast cancer-related information from Spanish clinical reports. They fine-tuned BERT-based and RoBERTa-based models on an annotated corpus and obtained high F1 results, with RoBERTa BNE performing best with percentage of 95.01 accuracy. Their results emphasize the importance of fine-tuning language models on biomedical datasets to improve structured data extraction from Spanish-language clinical texts for better oncology research and decision-making.

Nishioka *et al.* (2023) [113] developed an approach using deep learning to extract adverse event (AE) signals that impact patients' activities of daily living (ADL) from the blogs of Japanese breast cancer patients. In order to find AEs that restrict ADL, they trained the BERT, ELECTRA, and T5 models and analyzed and annotated 2,272 blog entries for AE-related references. T5 outperformed the other models in terms of accuracy, recall, and F1 scores (F1: 81.1 for all AEs). Pain, numbness, exhaustion, and nausea were the most often retrieved AEs. Their research demonstrates how NLP may be used to detect EAs and help medical practitioners provide improved patient care and prompt treatments.

Hu *et al.* [114] evaluated *BioBERT* for medical named entity recognition, their experiments with *BERT*, *ClinicalBERT*, *SciBERT*, and *BlueBERT* showed that *BioBERT* achieved the highest precision and F1 score. It is useful for clinical decision support and medical information extraction because of its biological pre-training, which improved domain-specific understanding. The study also highlighted the potential of *BioBERT* in intelligent medicine and disease prediction, and it addressed privacy issues and recommended merging specialized models for greater generalization.

When Chaudhury and Sau *et al.* (2023) [115] used *ViT* models in conjunction with *BERT* pre-training to categorize pictures of breast cancer, they were able to classify ultrasound and histological images with excellent accuracy. The combination of text-based and image-based models showed how multimodal techniques might increase the accuracy of diagnoses.

In the SemEval 2023 competition, Voloincu *et al.* (2023) [116] analyzed clinical trial data using *BioBERT* and *CNN models*, obtaining encouraging F1 scores for challenging NLP tasks. Their comparative study brought to light the advantages and disadvantages of different models for clinical inference tasks.

Watanabe *et al.* (2022) [117] used a *multilabel BERT classifier* on social media postings to detect treatment and financial problems among breast cancer patients, therefore facilitating prompt social support. The importance of *NLP* in comprehending and addressing the psychological components of patient care was demonstrated by this creative method.

Mutinda *et al.* [118] developed a Named Entity Recognition (NER) system based on BERT to automatically extract data for a meta-analysis from randomized controlled trials (RCTs) on breast cancer. They successfully identified the main PICO (Participants, Intervention, Control, and Outcomes) components from 1011 PubMed

abstracts (F1 score > 0.80). Their work shows how NLP can be used to automate meta-analyses, but further progress is needed to achieve greater statistical accuracy.

Choi *et al.* [119] evaluated the efficiency, cost, and accuracy of using LLMs to extract clinical factors from breast cancer reports. The LLM method achieved an accuracy of 87.7, with 98.2 for the detection of lymphovascular invasion. Their study highlights the potential of LLMs to automate the extraction of clinical data, reducing time and costs in breast cancer research.

In order to extract medical entities associated with breast cancer, recent studies took so many initiatives, like AI and NLP models. CancerBERT was introduced by Zhou et al. [111], who demonstrated enhanced NER performance. Solarte-Pabón et al. [112] refined transformation models for Spanish clinical reports, while Nishioka et al. [113] identified adverse events in Japanese patient blogs using deep learning. Multimodal techniques were used in other studies, such as Chaudhury and Sau et al. [115], and Voloincu et al. [116] used BioBERT to analyze clinical trial data.

3.2.1 Benchmark Evaluation for NER task

In addition to the previous work, we have added extra papers related to benchmark evaluation campaigns. A number of cutting-edge models have proven to perform well on biomedical NER benchmarks. BioBERT, for example, obtained an F1-score of 84.02 on the BC5CDR corpus and 89.04 on the NCBI Disease dataset [41]. With an F1-score of 89.72 on the NCBI dataset, PubMedBERT further enhanced performance [45], and BlueBERT has demonstrated efficacy in a variety of biomedical NLP tasks [44]. For biomedical applications, these models demonstrate the efficacy of domain-specific pretraining.

A variety of benchmark evaluation campaigns—most notably from CLEF and TREC—alongside numerous peer-reviewed studies, have significantly contributed to the advancement of biomedical NLP systems. These initiatives, often overlooked, provide shared datasets and well-defined tasks that promote reproducibility and enable rigorous comparative evaluations.

One of the earliest efforts in clinical named entity recognition (NER) that closely aligns with our multi-entity annotation framework is the i2b2/VA 2010 shared task, which focused on extracting key clinical concepts such as age, gender, medications, and diagnoses from de-identified medical records by Uzuner et al. [101]. Similarly aligned with our emphasis on capturing symptom–disease–medication associations, [102] offered a curated dataset for recognizing chemical and disease entities and identifying their co-mentions in biomedical abstracts. Introduced in later studies, the BC5CDR corpus has become a commonly used standard for named entity recognition (NER) of chemicals and diseases. It offers data with a BIO label that closely resembles our annotation format in both structure and intent [103].

[104] made a particularly relevant contribution by challenging participants to map disease and disorder mentions in clinical notes to standardized UMLS concepts. Its emphasis on span-level annotation and reliance on established clinical terminologies parallels our approach—especially in recognizing domain-specific entities such as cancer stage and treatment dosage. Further supporting our methodology, the [105] shared task advanced biomedical NER by introducing nested entities and emphasizing the complexity of real-world clinical text. Collectively, these initiatives offer a complementary backdrop to our study, which employs a diverse suite of transformer-based models and carefully curated data to extract entities specific to breast cancer.

Additionally, shared task campaigns such as TREC and CLEF have made significant contributions, laying the groundwork for biomedical Named Entity Recognition (NER). For instance, it [106] concentrated on extracting disease and gene-related information for treatment recommendations, while [107] concentrated on extracting disorders and medications from clinical texts. Additional initiatives, such as [108], highlighted the difficulties in identifying mental health indicators and focused on early risk detection using social media data. Similarly, disease mention annotations and normalization benchmarks were provided by [109]. The foundation for clinical NER benchmarking has been established by these efforts.

Compared to the previously mentioned campaigns, which primarily focused on broader or general clinical entity categories, our work introduces a fine-grained, domain-specific annotation schema tailored to the breast cancer domain. We annotated a diverse range of entities—*AGE, GENDER, SYMPTOMS, MEDICATION, DOSE, DISEASE, and CANCER STAGE*—many of which are underrepresented in standard benchmarks. To ensure quality and consistency, we assessed inter-annotator agreement (IAA) and reported detailed statistical summaries across training, validation, and test splits. Our methodology builds upon the foundation of prior shared tasks but advances it by leveraging transformer-based models like *BioBERT, ClinicalBERT, and BlueBERT*, along with an oncology-focused corpus and evaluation strategy specifically designed for breast cancer clinical narratives.

Following recent advancements in multilingual biomedical natural language processing, 'DrBenchmark' by [110] present a comprehensive evaluation benchmark specifically designed for the French biomedical field. It covers a wide range of tasks, such as document classification, named entity recognition, and question answering. Even though our framework was created for clinical data in the English language, the benchmark's structured design and domain-specific adaptation highlight how crucial task-specific evaluation is in biomedical settings. This complements our work and predicts future directions, including possible multilingual extensions, in our opinion.

Novelty of This Part. The studies collected through the systematic literature review (SLR) demonstrate that transformer-based models have significantly advanced clinical data extraction and classification. However, many prior studies focus on evaluating individual domain-specific BERT variants on standard biomedical datasets. In contrast, this work adopts a comparative evaluation strategy by systematically assessing both the pre-trained and fine-tuned versions of multiple domain-specific BERT models within a unified experimental framework. Furthermore, the evaluation uses a fine-grained annotation scheme for clinical narratives, enabling the extraction and assessment of specialized entities related to cancer diagnosis and treatment that are often underrepresented in existing biomedical NER benchmarks.

Takeaway. Building on existing biomedical NER tools, models, and benchmark initiatives, this study focuses on extracting fine-grained clinical entities (AGE, GENDER, SYMPTOMS, MEDICATION, DOSE, DISEASE, and CANCER STAGE) from clinical narratives. To achieve this, we evaluate multiple domain-specialized BERT variants using a consistent annotation scheme and a standardized performance evaluation protocol.

3.3 Knowledge Graphs in Biomedical Informatics

To understand state-of-the-art research relevant to knowledge graphs (KGs) and therapy recommendations, we performed a focused literature survey combining automated database searches and manual screening within a defined publication window.

LISTING 3.2: The Scopus advanced string search

```
TITLE-ABS (("Clinical Cancer Data" OR "Cancer Text Data" OR
   ↪  "Cancer Reports" OR "Medical Data" OR "Patient Data"
   ↪  OR "Medical Records" OR "Patient data" OR "
   ↪ Electronic Health Record")
AND ("Knowledge Graph*" OR "Graph Pattern" OR "Graph
   ↪ Pattern Analysis" OR "Graph Neural Network*" OR "GNN
   ↪ ")
AND ("Drug Interaction" OR "Medicine Recommendation System"
   ↪  OR "* Medicine Recommend* System*" OR "Drug
   ↪ Recommend* System*" OR "Therapy Recommend*" OR "
   ↪ Recommender System" OR "Drug-drug Interactions"))
AND PUBYEAR > 2015 AND PUBYEAR < 2025
```

After executing this string query in *Scopus advanced search*[2], we retrieved a total of 24 papers, shown in Listing 3.2. However, some of these papers were not directly linked to *"Pattern Matching"* but were relevant to *"Medicine Recommendation"* or *"AI techniques"*. From this collection, we carefully selected these relevant papers to analyze research gaps in the state-of-the-art, aiming to contribute towards advancing medical care and providing more efficient support for healthcare professionals.

Executing this query in Scopus returned 24 papers. A subset were not directly tied to *pattern matching* but were relevant to medication recommendation or broader AI techniques. We retained the most pertinent items for gap analysis and synthesis.

The studies in Table C.2 highlight a wide variety of approaches in biomedical informatics, especially in how knowledge graphs are applied to integrate Electronic Health Records (EHRs), biomedical ontologies, and drug–drug interaction databases. These approaches range from static graph modeling to dynamic, patient-specific graph construction, providing different levels of adaptability in clinical decision-making. However, as discussed in Section 3.4, most current systems do not fully exploit reinforcement learning or multi-objective optimization to adapt recommendations in real time.

Biomedical KGs and EHR integration. Biomedical KGs organize heterogeneous clinical knowledge—patients, diagnoses, procedures, labs, medications, and drug–drug interactions (DDIs)—into a graph that supports inference and recommendation. In practice, Electronic Health Records (EHRs) [120] are linked to external resources (e.g., DrugBank, ICD, UMLS/SNOMED) to form patient-specific subgraphs that capture temporal trajectories and therapy contexts. Graph representation learning then propagates information across related entities to support downstream tasks such as safety-aware medication recommendation, adverse event prediction, or cohort discovery.

[2] https://www.scopus.com/

GNNs for relational reasoning. A common thread in recent systems is the use of Graph Neural Networks (GNNs) to encode multi-relational structures. Studies such as [121, 122] augment patient graphs with adversarial/synergistic DDI edges to promote safer combinations, while [123] explicitly model both harmful and beneficial interactions to optimize therapeutic efficacy. Other lines incorporate attention, Transformers, and multimodal fusion (structured codes + clinical notes) to personalize predictions [124, 125]. Although these approaches substantially improve representational power, many are *static* (train–then–predict) and do not adapt policies as patient states evolve.

Limitations motivating our design. Most MKG-driven systems depend on preconstructed graphs and do not update edges/weights with new clinical evidence during deployment; safety checks (e.g., DDI filtering) are often applied post hoc rather than optimized jointly with efficacy. These observations motivate a dynamic, decision-oriented formulation that integrates KG reasoning with sequential decision-making —paving the way for the reinforcement-learning framework described next.

3.4 Reinforcement Learning in Therapy Recommendation

Medication recommendation is naturally a sequential decision problem: at each visit, the system must choose therapies that balance efficacy, safety (e.g., DDI risk), and personalization, while adapting to changes in a patient's condition over time. Several recent lines of work inch toward this goal. For example, time-aware or sequence-aware models like CompNet, MERITS, and DGCL introduce temporal encoders, neural ODEs, or contrastive objectives to leverage longitudinal data; however, most do not fully integrate *multi-objective* optimization (efficacy, safety, adaptability) into a unified reward design. Other systems—REST and FastRx among them—improve efficiency but still treat recommendation largely as classification or sequence labeling, rather than as policy learning.

Building on these insights, the final module of the MetaGRO framework introduces **RLGMO**, a reinforcement learning–based, graph-driven multi-objective model designed for personalized medication recommendation. RL-GMO uses (i) patient KGs (from EHRs and external ontologies) for context-aware state encoding; (ii) a multi-objective reward that internalizes safety (DDI control), efficacy (clinical improvement), and adaptability (robustness to evolving states); and (iii) a policy-learning module that continuously refines decisions as new evidence accrues.

Representative systems and comparison. Table 3.4 contrasts RLGMO against notable baselines referenced in our review (graph-based, safety-aware, temporal, and reinforcement components).

Evidence base synthesized into the sections Recent studies emphasize:

(a) *KG/GNN foundations* for representing patients, diagnoses, medications, and DDIs [121, 122, 123]; **(b)** *Transformer/attention* and multimodal fusion to encode rich context [124, 125]; and **(c)** early *RL/sequence-aware* formulations (e.g., CompNet, MERITS, DGCL) that begin to leverage trajectories and irregular time series. Yet, most systems still separate safety checks from the core optimization loop and/or do not adapt policies online. RL-GMO addresses these gaps by jointly optimizing safety and efficacy inside the reward while using graph-driven state representations to remain clinically grounded and explainable.

TABLE 3.4: Comparison of RLGMO with related medication recommendation methods

Feature	RLGMO (ours)	CompNet	REST	DGCL	FastRx
Reinforcement Learning	Yes	Yes	Yes	No	No
Graph-based Learning	Yes	Yes	Yes	Yes	Yes
DDI Awareness	Yes	Yes	Yes	Yes	Yes
Personalization	Yes	Yes	No	Yes	Yes
Reward Optimization	Yes (multi-objective)	Limited	Yes	No	No
Temporal Modeling	Yes (policy adapts)	No	No	No	Yes

Context from NER-driven oncology curation (bridging sections). When the clinical text provides sufficient information, the structured entities extracted in the earlier NER stage—such as symptoms, diseases, medications, doses, and cancer stage—can be used to populate patient-specific knowledge graphs and define the state representation for the RL model. This creates a coherent pipeline that links the upstream clinical NER process with downstream decision-support mechanisms, following the flow: *NER* $\rightarrow$ *KG construction* $\rightarrow$ *RL policy learning*. Such an integrated pathway enables personalized, safe, and adaptive therapy recommendations.

Novelty of This Part. Previous studies using knowledge graphs and graph convolutional networks (GCNs) have improved the representation of clinical data, while sequence-aware models have enhanced contextual understanding in medical decision support. However, most existing approaches rely on static learning frameworks and typically optimize a single objective. In contrast, **RLGMO** integrates graph-based clinical knowledge with reinforcement learning and a multi-objective reward design. This approach enables adaptive, safety-aware, and personalized therapy optimization within dynamic clinical environments.

Takeaway. Building on advances in knowledge graph modeling and reinforcement learning for clinical decision support, this work proposes the RLGMO framework to analyze patient histories and optimize therapy recommendations through graph-based contextual learning and a multi-objective reward mechanism.

Overall, many previous studies in clinical natural language processing focus on individual tasks such as entity extraction or text classification. In contrast, the proposed **MetaGRO** framework adopts an integrated approach that combines transformer-based entity extraction, knowledge graph construction, and AI-driven analysis to transform unstructured clinical narratives into structured medical knowledge. The framework captures clinically relevant entities related to cancer diagnosis and treatment and represents them within a knowledge graph, enabling connections between patient histories, clinical conditions, and treatment information. This representation supports the exploration of relationships among similar patient cases and provides insights for therapy analysis and optimization. In addition, MetaGRO incorporates FAIR-aligned data management by storing structured outputs within a research infrastructure that enables transparent sharing, visualization, and reuse of results.

Chapter 4

Entity Extraction from Clinical Textual Data

This chapter is organized into a set of sections that together describe the complete process of extracting clinical entities from textual data. Section 4.1 outlines the overall methodology followed in designing the entity extraction pipeline.

Section 4.2 then describes the medical entity extraction pipeline using pre-trained BERT-based models, while Section 4.3 details the fine-tuning environment and the experimental setup employed to adapt these models for clinical NER. Section 4.4 explains the evaluation procedures and performance metrics used in this work and outlines how these evaluations address the research questions and the main objectives of the study. Section 4.5 presents the comparative results across the different models, and Section 4.6 discusses the NER performance and entity-level accuracy.

Finally, Section 4.7 provides a broader discussion of the findings, and Section 4.8 concludes the chapter by summarizing the key insights and highlighting directions for further work.

4.1 Methodology

This section provides an overview of the proposed methodology we implemented to select the best BERT-based model for the medical entity extraction task. It is depicted in Figure 4.1, and it consists of multiple phases we describe below. Phase (1) focuses on data cleaning and preprocessing to ensure high-quality input for subsequent phases [(3) and (4)] (detailed in Section 4.1.1). Phase (2) describes the manual annotation process to generate a labeled dataset in CSV format in Section 4.1.2, which is then converted into BIO tagging format. This tagged data serves as the ground truth corpus for both model training and performance evaluation in Phases [(3) and (5)], respectively. Phase (3) covers the extraction of medical entities using these fine-tuned models (Section 4.1.3). In Phase (4), we present the fine-tuning of pre-trained BERT-based models aligned to the annotated dataset (detailed in Section 4.1.4). Finally, Phase (5) provides a performance evaluation and comparative analysis to identify the most effective BERT-based model for our selection task (in Section 4.1.5).

In our study, data comes from a research-based medical database[1] that contains unannotated clinical text collected from the University of California, San Francisco (UCSF) Information Commons in the period from 2012 to 2022, publicly available clinical reports with a data use agreement, and need a valid CITI certificate[2].

[1] `https://physionet.org/`

[2] `http://physionet.org/about/citi-course/`

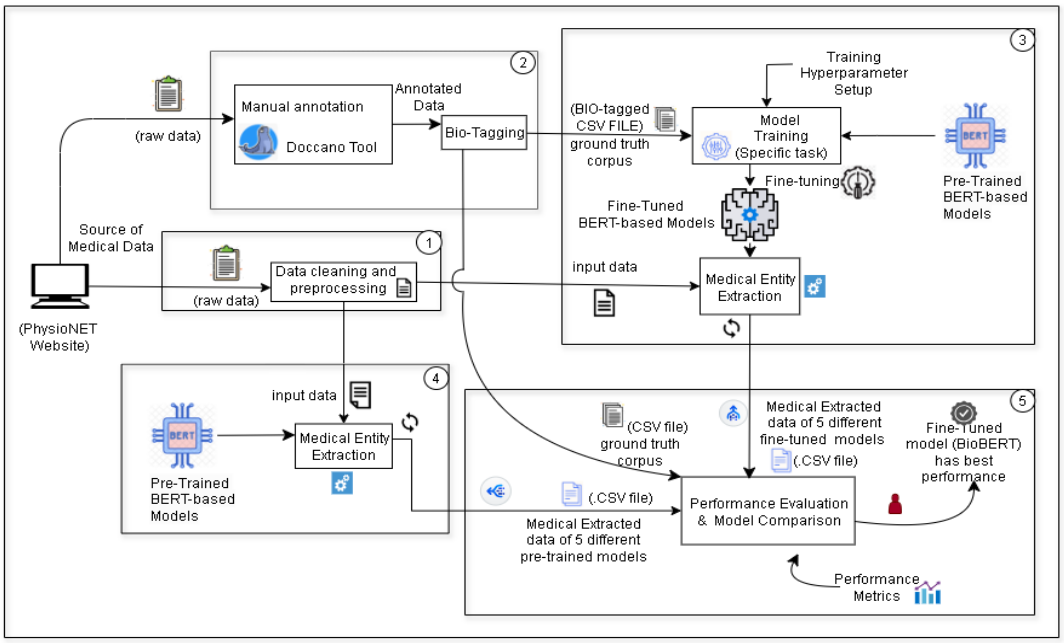


FIGURE 4.1: The workflow implemented to select the best BERT-based model for the clinical entity extraction task.

Instead, the considered BERT-based models come from the analysis of the state of the art made in Section 2.1.3: *BioBERT*, *BioClinicalBERT*, *PubMedBERT*, *BlueBERT*, and *RoBERTa*. These models are already pre-trained on large biomedical or clinical datasets shown in Table 2.1, such as PubMed, PMC, and MIMIC-III, except for *RoBERTa*, which is trained on general domain corpora like Common Crawl and Books. Despite not being specialized in biomedical texts, *RoBERTa* is included as a baseline model to evaluate its generalization ability and to compare how domain-specific fine-tuning improves its performance on medical entity extraction tasks.

4.1.1 Phase 1: Data Description with cleaning and Preprocessing

In this phase, we present the introduction of clinical datasets used in this study and explain how the raw text data was prepared for the further process.

Data description

A dataset consisting of 40 cancer patient data samples was curated by Sushil et al. [126] from the University of California, San Francisco (UCSF) Information Commons in the period from 2012 to 2022. All data were deidentified for patient information, and dates were offset. Only those patients with complete staging information available, documented disease progression, and oncology notes were selected.

Some gene symbols, clinical trial names, and cancer stages that were inappropriately redacted during automated processing were manually restored under UCSF Institutional Review Board approvals *18-25163* and *21-35084*. We selected breast cancer because it is often a treatable disease with extensive biomarker and genetic testing and integrative treatment plans incorporating radiation, surgery, and medical oncology.

All narrative text sections were annotated—excluding copy-forward reports by oncology fellows and a medical student—after which 40 unannotated clinical notes

were made available on the PhysioNet website[3]. The 'coral' folder contains the 'annotated' subfolder, which is for breast cancer notes and contains '.txt' and '.ann' files with all the annotations. In summary, this 'annotated' section contains a total of 9,028 entities, 9,986 modifiers, and 5,312 relationships in total, hence a big dataset to work on and study concerning breast cancer.

Use of Auxiliary Pancreatic Cancer Data

Our research primarily focuses on breast cancer clinical data, which serves as the main domain for model fine-tuning and evaluation. To address the limitation of sample size, a small set of pancreatic cancer records was additionally used during the fine-tuning phase. These supplementary data were included solely to improve model generalization and training stability, without shifting the core focus away from breast cancer. Incorporating pancreatic cancer data helped the model capture broader medical language patterns and enhance contextual understanding, while all analyses and results presented in this study remain centered on breast cancer.

Here we prepare raw clinical text data before performing the entity extraction task. This cleaned and preprocessed data serves two key purposes: *(1)* as input for fine-tuned versions of the BERT-based models (in phase 3), and *(2)* as input for pre-trained BERT-based models (in phase 4). Specifically, we applied the following preprocessing steps:

Text Cleaning The clinical text is first converted to lowercase. Next, non-informative characters like repeated special symbols (such as "*", "!", "@", and "$"), irregular punctuation patterns, and embedded noise like additional white spaces or formatting artifacts are eliminated. The goal of this stage is to minimize input clutter while maintaining the essential semantic information needed for entity detection. To ensure that the models consistently processed natural, entire sentences, it is particularly important to note that this cleaning primarily focused on formatting noise and disruptive symbols, leaving the original sentence structure and clinically significant elements (such as verbs and function tokens) intact.

We realize that lowercasing may not always work with the pretraining configurations of some encoders (for example, RoBERTa is pretrained on cased text). Though cased models like RoBERTa likely suffered a minor drawback as a result of this decision, our pipeline consistently used lowercasing to simplify auxiliary preprocessing. Future studies must specifically compare cased vs. uncased preprocessing to see if model-specific casing procedures lead to noticeable improvements.

Noise Token Filtering and POS-Based Selection Rather than removing traditional stopwords as in classical information retrieval, we applied a targeted filtering approach only in auxiliary analyses (e.g., feature exploration, word cloud generation). Using the NLTK and SpaCy libraries, we retained linguistically relevant tokens—primarily nouns and adjectives—which are known to carry semantic weight in identifying clinical entities such as symptoms, diseases, and medications.

During this process, verbs and other functional tokens were excluded only for auxiliary preprocessing tasks, not for model training or evaluation. All BERT-based

[3] `https://physionet.org/content/curated-oncology-reports/1.0/`

models were trained and tested on full sentence inputs without token exclusion, ensuring that contextual information was preserved. We did not execute a specific experiment to quantify the impact of this filtering on performance, and future research should measure filtered and unfiltered pipelines to ascertain its useful applicability.

Splitting Long Text into Chunks Due to the 512-token input limit of BERT-based models (e.g., BioBERT, BlueBERT), we split long clinical notes into smaller, manageable chunks using the function `split_text_into_chunks`. This ensures the entire narrative is preserved and processed without truncating important clinical information.

Overall, the pre-processing was purposefully kept to a minimum to prevent altering the model's capabilities; entity extraction depended on the whole context of the original clinical text, and fundamental sequence information was maintained.

4.1.2 Phase 2: Manual Annotation

The raw clinical text data was meticulously annotated using the *Doccano* open-source tool, with contributions from two expert annotators specializing in the health field.

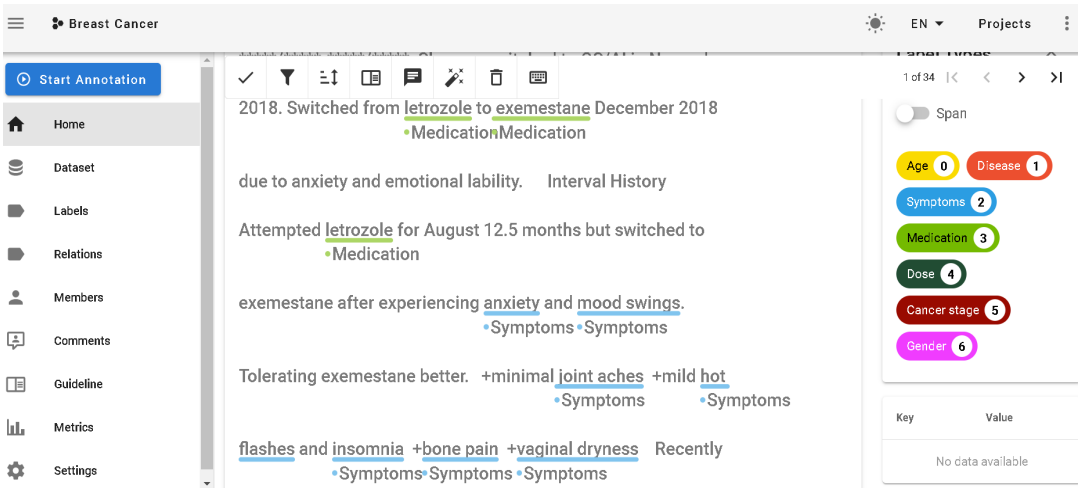


FIGURE 4.2: The manual annotation of mammary malignancy data using the Doccano tool.

This tool allows users to create and share projects with team members for collaborative annotation, enabling easy text labeling, as shown in Figure 4.2. To ensure the annotated corpus is reliable and suitable for establishing a ground truth, we evaluated the agreement between the annotators using Cohen's Kappa shown in Table 4.1. This metric provided a robust measure of consistency, confirming the corpus's suitability as a ground truth dataset.

Since the data was already anonymized and reviewed by medical professionals, no additional review of the text data was necessary. However, during the annotation process, every effort was made to ensure accuracy and adherence to guidelines. Cohen's Kappa value (κ) is a statistical measure that evaluates the level of agreement between two annotators while accounting for the agreement occurring by chance. The formula for Cohen's Kappa is [127]:

$$\kappa = \frac{P_o - P_e}{1 - P_e}$$

TABLE 4.1: Inter-Annotator Agreement (IAA) scores and Cohen's Kappa for various annotation tasks.

Annotation Task	Annotator 1 (%)	Annotator 2 (%)	Cohen's Kappa	Interpretation
Age Annotation	94%	93%	0.85	Almost Perfect Agreement
Gender Annotation	96%	95%	0.90	Almost Perfect Agreement
Disease Annotation	95%	94%	0.89	Almost Perfect Agreement
Symptom Annotation	89%	91%	0.76	Substantial Agreement
Medication Annotation	92%	88%	0.64	Substantial Agreement
Dose Annotation	85%	87%	0.57	Moderate Agreement
Medical History Annotation	84%	86%	0.56	Moderate Agreement
Cancer Stage Annotation	90%	92%	0.80	Substantial Agreement

Where P_o represents the **observed agreement**, which is the proportion of instances where the annotators agree, and P_e represents the **expected agreement**, which is the proportion of agreement expected by chance alone.

The observed agreement (P_o) is calculated as:

$$P_o = \frac{\text{Number of agreements between annotators}}{\text{Total number of instances}}$$

The expected agreement (P_e) is determined by summing the product of the marginal probabilities of each category assigned by the annotators:

$$P_e = \sum_{i=1}^{k} (P_{A,i} \times P_{B,i})$$

Here, $P_{A,i}$ and $P_{B,i}$ denote the proportions of instances where Annotator A and Annotator B, respectively, assigned category i, and k is the total number of categories.

The value of κ ranges from -1 to 1, **Where:** $\kappa = 1$ indicates perfect agreement, $\kappa = 0$ represents agreement equivalent to random chance, and $\kappa < 0$ signifies agreement worse than chance.

Cohen's Kappa is widely used for assessing the reliability of annotated data, ensuring that the annotations are consistent and suitable for downstream tasks. The interpretation of κ values ranges from **Poor Agreement** ($\kappa < 0.00$) to **Almost Perfect Agreement** ($0.81 \leq \kappa \leq 1.00$).

During the manual annotation phase, we defined seven distinct named entity classes to guide the annotation process. These classes are:

– AGE: Patient's age information in demographic, either numerical or descriptive.

– GENDER: Mentions the gender of a person; it is male or female.

– DISEASE: Identified medical conditions and diagnoses for the patients.

– SYMPTOMS: Clinical symptoms observed or reported based on the disease diagnosed.

– MEDICATION: Therapeutic drugs prescribed or referenced based on the symptoms of a particular disease.

– DOSE: Dosage information of the medications taken by the patients.

– CANCER STAGE: Cancer classification based on staging systems (e.g., Stage I–IV).

The importance of these entity types in clinical narratives pertaining to oncology and their applicability to later uses like clinical decision-making and treatment

recommendation systems led to their careful selection. The subsequent BIO-tagging procedure was built upon this systematic classification, which also directed the development of our annotation guidelines.

The output of the annotation and BIO tagging process is saved as a structured CSV file, where each row corresponds to a single token and includes BIO labels across the seven classes defined above. This format provides the token-level granularity needed for multi-entity fine-tuning and evaluation. For example, the sentence *"The patient was prescribed Tamoxifen for breast cancer."* may be represented in the BIO-tagged CSV format as follows:

```
Sentence_ID: 0
Token:  Tamoxifen - MEDICATION: B-MEDICATION
Token:  breast - DISEASE: B-DISEASE
Token:  cancer - DISEASE: I-DISEASE
```

After the annotation process, we carried out a thorough statistical analysis to improve the transparency and evaluability of our annotated dataset. Across the train, validation, and test splits, we calculated the frequency of each annotated class. In addition, we computed the total and average number of words and sentences in the raw text to understand the structural characteristics of the data prior to annotation. This preprocessing involved standardization (lowercasing, punctuation cleaning, and whitespace normalization), followed by whitespace-based tokenization to reflect the model's actual input format. These statistics are essential for assessing the diversity and distributional balance of the dataset, which directly impact model performance and generalizability. Table 4.2 summarizes the sentence and word counts from the original, unannotated raw text cancer data, while Table 4.3 presents the entity-level annotation distribution.

TABLE 4.2: Sentence and Word Statistics Across Dataset Splits.

Split	Total Sentences	Avg Sentences per File	Total Words	Avg. Words per File
Train	2963	109.74	72286	2677.26
Validation	772	110.29	18788	2684.00
Test	570	95.00	14600	2433.33

TABLE 4.3: Distribution of annotated entity types across dataset splits.

Entity Class	Count of Annotations	IAA
AGE	34	0.85
DISEASE	491	0.89
SYMPTOMS	1172	0.76
MEDICATION	592	0.64
DOSE	1090	0.57
GENDER	34	0.90
CANCER STAGE	24	0.80

The distribution of annotated entity classes for the training and validation sets, which make up 85% of the total annotated data, is shown in Table 4.3. To maintain

evaluation integrity, the remaining 15% was set aside as a held-out test set and excluded from the annotation statistics displayed here. Across the seven classes AGE, GENDER, CANCER STAGE, DISEASE, SYMPTOMS, MEDICATION, and DOSE, we manually annotated 34 clinical documents, yielding 3,437 entity mentions. In order to guarantee representative coverage across entity classes, the total 40 clinical datasets were then divided into 70% training, 15% validation, and 15% testing.

4.1.3 Phase 3: Medical Entity Extraction using fine-tuned BERT-based Models

To enhance medical entity recognition, we fine-tune five pre-trained BERT-based models, and each model is trained on the annotated dataset (Bio-tagged) prepared as described in Section 4.1.2. Each model is initialized with domain-specific weights and adapted further to our cancer-related dataset. This step aims to fine-tune the BERT-based models to the considered medical domain, improving their ability to extract key medical entities with higher accuracy. This fine-tuning is performed using the *Hugging Face Trainer API*, where pre-trained models are available, and helps these models understand domain-specific language, making them more effective in real-world applications during the ongoing training process.

The labeled dataset obtained from Phase (4.1.2) is used as the ground truth corpus for fine-tuning. This corpus includes clinical entities such as diseases, symptoms, medications, cancer stages, and other relevant medical information. The annotated data is formatted using the BIO (Beginning, Inside, Outside) tagging[4] scheme, ensuring a high-quality supervision ground truth corpus for training the models.

Initially, the manually annotated CSV file is preprocessed by converting it into the BIO-tagging format, a standard scheme in named entity recognition tasks that encodes the position of entity tokens within a text. This transformation facilitates effective token classification by the models. The fine-tuning process employs specific hyperparameters, detailed in Table 4.4, which have been optimized to enhance model performance.

TABLE 4.4: Hyperparameters used for training the models.

Hyperparameters	**Values**	**Hyperparameters**	**Values**
Random Seeds	42, 56, 101, 202, 303	Batch size buffer	256
Epochs	20	Discard oversize batches	True
Dropout	0.1	Learning rate scheduler	Warmup-linear
Optimizer	AdamW	Initial learning rate	5e-5
GPU allocator	PyTorch	Total training steps	3,000 (approx.)
Batch size	16–32	Warmup steps	300

The *max_length* parameter defines the maximum number of tokens that the model will accept in a single input sequence. It ensures that tokenized clinical text does not exceed the model's limit (typically 512 tokens for BERT variants) and triggers chunking when input exceeds this size. Then *Batch_size* parameter specifies the number of input samples processed together in a single pass through the model. A larger batch

[4]The BIO format is widely used in sequence labeling tasks, such as named entity recognition, to denote the beginning, inside, and outside of entities within any sentence. This scheme aids models in distinguishing entity boundaries effectively.

size increases processing speed but also requires more GPU memory. *Stride* controls the overlap between consecutive chunks when long texts are split. It ensures that entities near the chunk boundaries are not missed by maintaining token overlap across chunks. Also, *learning_rate* determines the size of the weight updates during fine-tuning. A smaller learning rate results in slower but more stable convergence, while a larger rate speeds up learning but risks unstable updates. This *Epoch* is the parameter that defines how many times the entire dataset is passed through the model during fine-tuning. More epochs allow for better learning but increase the risk of overfitting if set too high.

During the fine-tuning process, tokenized text-label pairs are provided to the models to adjust their parameters, enabling them to capture the domain-specific patterns more effectively. This targeted training adapts the models to enhance their ability to extract clinically relevant entities, such as diseases, symptoms, medications, and cancer stages, from unstructured medical text.

After fine-tuning, the models are used to extract entities from preprocessed clinical text data. This phase (4.1.3) is shown in Figure 4.1. The extraction process is carried out using the methods implemented in the GitHub repository. First, the clinical text is tokenized using the Hugging Face Transformers library (via `AutoTokenizer`) and passed into the fine-tuned model loaded with `AutoModelForTokenClassification`, which is suited for token classification tasks like Named Entity Recognition (NER). The model is trained to recognize entities based on the BIO tagging format, where each token is labeled as the beginning (B), inside (I), or outside (O) of a medical entity.

After generating predictions, special tokens such as `[CLS]`, `[SEP]`, and `[PAD]` are removed. Subword tokens marked with `##` (e.g., `'Tam'`, `'##ox'`, `'##ifen'`) are merged to reconstruct full medical terms (e.g., "Tamoxifen"). The final extracted entities—including diseases, symptoms, medications, doses, and cancer stages—are compiled and exported to CSV files for further evaluation and analysis, using the same BIO format as the annotated ground truth.

4.1.4 Phase 4: Medical Entity Extraction using Pre-trained BERT-based Models

It should be mentioned that in order to complete the evaluation, we additionally take into account the zero-shot (pre-trained model) condition in order to demonstrate the distinction between the model performance attained through direct inference and fine-tuning. Masked language models, such as BioBERT or PubMedBERT, have been shown to perform significantly lower in biomed NLP when fine-tuning is not performed. This is because these models need to be fine-tuned in order to achieve optimal performance.

This helps analyze the significance of fine-tuning for optimal clinical entity recognition in more depth, alongside Phase 4.1.3. In this phase, we applied medical entity extraction using the five BERT-based pre-trained models. The outcome of the pre-trained models will act as a point of comparison to view the performance of the models from phase three.

Each pre-trained model processes input data using tokenization via `AutoTokenizer`, followed by entity extraction through `AutoModelForTokenClassification` in inference mode. The clinical text is first cleaned and preprocessed using the generalized functions (e.g., `clean_text()` and `remove_negative_phrases()`) available in the GitHub repository to eliminate all unnecessary noises and non-informative phrases.

Due to the input length limitation of transformer models, the cleaned text is then divided into smaller overlapping segments (using `split_text_into_chunks()` function). This ensures each chunk fits within the model's maximum input length (512 tokens) constraints while retaining important contextual information across segment boundaries.

Bio-Tagging The input clinical text is first tokenized and passed through pre-trained models to generate token-level predictions. These predictions utilize the BIO (Beginning, Inside, Outside) tagging scheme, a standard method in named entity recognition [128] tasks that labels each token to indicate its position relative to an entity—*'B' denotes the beginning, 'I' inside, and 'O' outside*. The following example in the text box below shows how each token in a clinical sentence is aligned with its corresponding BIO label, visually representing the structure of the annotated dataset used for model training.

The predicted outputs are then post-processed to map each token to its corresponding medical entity, covering categories such as diseases, symptoms, medications, dosages, and cancer stages. Each token's predicted label is mapped from index to entity tag using the model's label schema (e.g., B-DISEASE, I-MEDICATION) and grouped accordingly.

The	patient	was	prescribed	Tamoxifen	for	breast	cancer	.
O	O	O	O	B-MEDICATION	O	B-DISEASE	I-DISEASE	O

Subword tokens are merged to reconstruct complete entities, while special tokens, as discussed in Section 4.1.3, are removed to ensure clean and accurate entity lists. Finally, the extracted entities are exported into CSV files, which are used as inputs for performance evaluation and comparison of the results of both pre-trained and fine-tuned models, as described in Section 4.1.5.

For every backbone model (BioBERT, BioClinicalBERT, PubMedBERT, BlueBERT, and RoBERTa), we appended a common token classification head consistent with our annotation scheme (AGE, GENDER, DISEASE, SYMPTOMS, MEDICATION, DOSE, CANCER STAGE). It was randomly initialized for the zero-shot scenario and fine-tuned. The reason for providing a zero-shot baseline is two-fold: *(i)* it shows the shortcomings of using pre-trained biomedical encoders alone without adaptation for specific tasks, and *(ii)* it sets the value added by fine-tuning for clinically relevant entity extraction. In practice, institutions might first try to deploy off-the-shelf biomedical models without annotation tools; thus, showing both conditions deemphasizes the need for fine-tuning when accurate entity extraction is essential.

4.1.5 Phase 5: Performance Evaluation and Model Comparison

To identify the most effective model for medical entity extraction, we conduct a performance evaluation of both pre-trained and fine-tuned models using the ground truth corpus prepared in Section 4.1.2. The evaluation is performed on the extracted entities generated results in Sections 4.1.4 and 4.1.3. The evaluation process uses standard metrics such as *accuracy*, *precision*, *recall*, and *F1-score* described in Paper [129], These metrics are calculated by comparing the extracted entities (diseases, symptoms, medications, doses, and cancer stages) from both pre-trained and fine-tuned models with the annotated ground truth dataset prepared in Phase (4.1.2). We report token-level F1 (macro-averaged) following standard NER practice, which better reflects entity-level extraction performance in imbalanced datasets.

To carry on the performance evaluation and model comparison, each model's extracted entities are processed to determine the selected evaluation metrics (using *evaluate_performance()* function). The comparison is carried out for all five models: *BioBERT*, *BioClinicalBERT*, *PubMedBERT*, *BlueBERT*, and *RoBERTa*. Both pre-trained and fine-tuned outputs are evaluated separately to highlight the performance improvements gained through fine-tuning.

Comparing the metrics results, it is evident that the best-performing model is the fine-tuned BioBERT. It achieves the highest *accuracy* and balanced *F1-score and recall* values. The result has been shown in Section 4.5.

4.2 Medical Entity Extraction Pipeline using Pre-trained BERT-based models

4.2.1 Entity Extraction using BioBERT (Zero-Shot Setting)

BioBERT [41] (Bidirectional Encoder Representations from Transformers for Biomedical Text Mining) is a domain-specific variant of BERT that has been pre-trained on large-scale biomedical corpora, including PubMed abstracts and PMC full-text articles. Its adaptation to biomedical language enables superior performance in extracting domain-relevant entities such as diseases, symptoms, and medications compared to general-purpose models. Leveraging BioBERT's zero-shot capability, the following pipeline outlines our entity extraction process, which identifies key biomedical entities directly from unannotated clinical text without additional fine-tuning.

Zero-Shot Entity Extraction Process using Pre-trained BioBERT Models

In algorithm 1, the zero-shot entity extraction approach with BioBERT eliminates the need for additional supervised training data. By using BioBERT's domain-specific pre-training on large-scale biomedical corpora, the model can recognize key biomedical entities directly from raw text. The detailed process is outlined below.

1. Input and Model Initialization

Pre-trained Model	`alvaroalon2/biobert_diseases_ner`
Framework	Hugging Face Transformers (Token Classification)
Input Data	Raw clinical or patient text files
Mode	Zero-shot (no fine-tuning)
Seeds	{42, 56, 101, 202, 303} for reproducibility
Tokenizer	BioBERT's WordPiece tokenizer
Output	Token-level predictions for each entity label

2. Pre-processing and Tokenization

Text Cleaning	Convert text to lowercase, remove special characters except `[a-z, A-Z, 0-9 /.,-]`, and collapse extra spaces.
Chunking	Split text into overlapping segments of 512 tokens with an overlap of 50 tokens to maintain context.
Tokenization	Apply BioBERT's WordPiece tokenizer to generate input IDs and attention masks.

3. Model Inference (Zero-Shot Prediction)

– Each chunk is passed through the pre-trained BioBERT model.
– The model outputs logits for each token.
– The highest-scoring label is selected using `argmax`.
– Tokens labeled as `[CLS]` or `[SEP]` are ignored.
– Predictions correspond to predefined entity label IDs (e.g., Disease or Symptom).

4. Post-processing of Predictions

– Merge subword tokens starting with "##" to form complete words.
– Combine contiguous tokens with the same label to form entity phrases (e.g., "breast" + "cancer").
– Remove duplicates and collect unique entity terms for each file.

5. Entity-wise Extraction Rules Table 4.5 summarizes the entity-extraction terms identified from unstructured clinical textual data using BioBERT and other BERT-family models.

TABLE 4.5: Entity types and their extraction rules in the zero-shot BioBERT pipeline.

Entity Type	Extraction Logic / Description	Example Outputs
Age	Regular expressions identify numeric patterns followed by terms like "years old," "y.o.," or "age."	45 years old; 60 y.o.
Gender	Search for gender-related terms (male, female, man, woman) appearing near demographic context.	male; female
Disease	Model-predicted tokens labeled as disease entities, including medical and cancer-related conditions.	breast carcinoma; lung metastasis
Symptoms	Model-predicted tokens representing observable or reported symptoms.	fatigue; chest pain; weight loss
Medication	Combined rule-based and model predictions; regex detects "DrugName + number + unit."	Tamoxifen 20 mg; Letrozole 2.5 mg
Dose	Regular expressions extract numeric dosages with measurement units.	500 mg; 2.5 ml
Cancer Stage	Pattern recognition of cancer staging (Stage I–IV, TNM classification).	Stage II; Stage IVB; T2N1M0

6. Output Generation

– Extracted entities are saved in a structured CSV file.
– Each record includes: `Filename`, `Age`, `Gender`, `Disease`, `Symptoms`, `Medication`, `Dose`, and `Cancer Stage`.
– Each run is tagged with the corresponding seed value for reproducibility.

7. Key Characteristics of Zero-Shot Extraction

No Fine-Tuning	Uses BioBERT's existing biomedical knowledge directly.
Domain Adaptation	Pre-training on PubMed and PMC enhances understanding of medical context.
Label-Free Recognition	Detects entities via semantic and contextual similarity rather than task-specific labels.
Cross-Domain Reusability	Performs well across unseen medical datasets without retraining.

The zero-shot entity extraction algorithm 1 for the BioBERT model works as follows. We begin with a set of random seeds $\mathcal{S}$ and two input directories where all cancer text files are stored $\mathcal{D} =$ /breastca, /pdac, and we use the pre-trained BioBERT disease NER model m. First, the tokenizer T is loaded using the model m. Then, for every seed $s \in \mathcal{S}$, the seed value is fixed using SETSEED(s) to make the results reproducible, and a fresh model instance M is loaded from m. For each seed, the procedure PROCESSDIRECTORY$(\mathcal{D}, T, M, s)$ goes through every directory $d \in \mathcal{D}$ and processes all the `.txt` files. For every file f, the text is read, cleaned, and passed to EXTRACTENTITIES(text, T, M), which returns a structured dictionary of extracted information. The results are collected into a list $\mathcal{R}$ and finally written into an output CSV file corresponding to the seed.

Inside EXTRACTENTITIES, the input text is first cleaned to produce u. Diseases and symptoms are extracted using the function EXTRACTDISEASESSYMPTOMS(u, T, M), which returns two sets $(\mathcal{D}^\star, \mathcal{S}^\star)$.

Since the BioBERT model can handle only 512 tokens at a time, the text u is split into overlapping chunks using SPLITCHUNKS. Each chunk is tokenized using T and passed through the model M to obtain logits. The predicted labels $\hat{y} = \arg\max(\text{logits})$ are mapped back to tokens, and then CLEANTOKENS collects all spans labelled as Disease (label = 1) and Symptom (label = 2). In addition to these model-based entities, other clinical details—such as medication names, doses, age, gender, and cancer stage—are extracted using simple regex-based functions like EXTRACTMEDICATIONS, EXTRACTAGE, EXTRACTGENDER, EXTRACTCANCERSTAGE, and EXTRACTDOSES. The final output returned is a dictionary containing Age, Gender, Cancer stage, Disease = uniq$(\mathcal{D}^\star)$, Symptoms = uniq$(\mathcal{S}^\star)$, Medication, Dose. This complete process is repeated for every seed, making the pipeline robust and easy to apply for zero-shot entity extraction from raw clinical text.

4.2.2 Entity Extraction using other BERT-based models (Zero-Shot Setting)

To ensure comprehensive evaluation and comparison, the same zero-shot entity extraction procedure used for BioBERT was applied to four additional Transformer-based models: BioClinicalBERT, PubMedBERT, BlueBERT, and RoBERTa. Each model shares the same architecture and processing workflow but differs in its pre-training domain and corpus. BioClinicalBERT is trained on clinical notes (MIMIC-III dataset), PubMedBERT on biomedical abstracts, BlueBERT on a combination of PubMed and clinical text, and RoBERTa serves as a general-domain baseline model. All models follow the identical entity extraction pipeline, as summarized in Algorithm 2.

This algorithm 2 simply runs the same zero-shot extraction pipeline across four different pre-trained NER models: BioClinicalBERT, PubMedBERT, BlueBERT, and RoBERTa. For each model m, the corresponding tokenizer T and token-classification model M are loaded, and the pipeline is executed for all seeds $s \in \mathcal{S}$ to keep the

results reproducible. The procedure PROCESSDIRECTORY then goes through every `.txt` file inside the cancer files of the directories $\mathcal{D}$, reads the text, and extracts entities using EXTRACTENTITIES, which internally applies model-based predictions for diseases and symptoms and regex (regular expression)-based rules for the remaining attributes. The extracted entities, along with the filename and seed, are collected in $\mathcal{R}$ and finally written into a CSV file named according to the model and seed. In short, the algorithm ensures that all four models are evaluated in exactly the same way, allowing a fair zero-shot comparison across models.

4.3 Fine-Tuning Environment and Experimental Setup

4.3.1 Fine-tuning BioBERT for clinical entity extraction

Task Definition

The objective of this experiment is to fine-tune the pre-trained **BioBERT** model for **clinical named entity recognition (NER)** using the BIO tagging format (e.g., `B-DISEASE`, `I-DISEASE`, `O`). The task involves assigning a label to each token within a clinical text sequence to identify disease names, symptoms, medications, doses, stages, gender, and age-related entities.

The BIO tagging scheme is adopted because it supports multi-token entities (e.g., "breast cancer" $\rightarrow$ `B-DISEASE I-DISEASE`) and helps preserve entity boundaries during model evaluation using entity-level F1 metrics.

Software and Hardware Environment

All fine-tuning experiments were conducted using the **Kalifano Cloud Computing Infrastructure** at the University of L'Aquila (UNIVAQ). This platform provides high-performance computing (HPC) resources optimized for research in artificial intelligence and data science.

Kalifano offers a scalable virtualized environment with NVIDIA GPU support and a pre-configured software stack suitable for deep learning workloads. The infrastructure is widely used within UNIVAQ for training large language models, running reinforcement learning simulations, and performing high-throughput biomedical data analysis.

The model was fine-tuned using the following environment specifications:

Platform	Kalifano Cloud (UNIVAQ HPC)
Operating System	Ubuntu 22.04 LTS
GPU	NVIDIA RTX A6000 / Tesla V100 (depending on allocation)
CPU	16-core Intel Xeon Silver / AMD EPYC
Memory	64–128 GB RAM
CUDA Version	12.1
Frameworks and Libraries	`PyTorch` – for model training and optimization `Transformers` (Hugging Face) – for model loading, tokenization, and training utilities `Datasets` – for managing and processing BIO-formatted text data `SeqEval` – for computing entity-level Precision, Recall, and F1-Score `Scikit-learn` – for auxiliary metric computation `NumPy` and `Pandas` – for data handling and aggregation

The required libraries can be installed using the following command:

```
pip install torch torchvision torchaudio transformers datasets

seqeval scikit-learn numpy pandas
```

Pre-trained Model and Tokenizer

The base model used for fine-tuning is **BioBERT** (`alvaroalon2/biobert_diseases_ner`), which is a domain-specific BERT model pre-trained on PubMed abstracts and PMC full-text articles. The tokenizer used is a **WordPiece tokenizer**, initialized as:

```
AutoTokenizer.from_pretrained("alvaroalon2/biobert_diseases_ner")
```

Data Format and Preprocessing

Each input example consists of a sequence of tokens with corresponding BIO labels. The dataset follows the format:

```
{
  "tokens": ["The", "patient", "has", "breast", "cancer", "."],
  "labels": ["O", "O", "O", "B-DISEASE", "I-DISEASE", "O"]
}
```

During preprocessing:

- Tokens are split into subwords using the WordPiece tokenizer.
- Labels are aligned with subwords so that only the first subword receives the label, while subsequent subwords are assigned the value `-100`, indicating that they are ignored during loss computation.
- The dataset is split into training and validation sets (70:15 ratio) using fixed random seeds for reproducibility.

Training Configuration

The fine-tuning was conducted using the `Trainer` API from Hugging Face with the following hyperparameters:

TABLE 4.6: Hyperparameters Used in BioBERT Fine-Tuning

Parameter	Symbol	Value / Description
Epochs	E	10 (range: 20)
Batch Size	B	16–32 (based on GPU memory)
Max Sequence Length	$L_{\max}$	128–256 tokens
Learning Rate	η	5×10^{-5}
Weight Decay	λ	0.01
Warmup Ratio	ρ	0.1 (linear scheduler with warmup)
Random Seeds	$\mathcal{S}$	{42, 56, 101, 202, 303}
Optimizer	–	AdamW
Loss Function	–	Cross-Entropy (ignores masked tokens)
Evaluation Metric	–	SeqEval entity-level F1, Precision, Recall

Evaluation Metrics

Performance is measured using entity-level metrics rather than token-level metrics to ensure that multi-token entities are evaluated as single units. The following metrics are computed:

- **Precision:** Correctly predicted entities / Total predicted entities
- **Recall:** Correctly predicted entities / Total true entities
- **F1-Score:** Harmonic mean of Precision and Recall
- **Accuracy:** Token-level agreement (optional)

The final scores are reported as the mean $\pm$ standard deviation (SD) across multiple random seeds:

$$\text{Metric}_{avg} = \frac{1}{|\mathcal{S}|} \sum_{s \in \mathcal{S}} \text{Metric}(s)$$

Reproducibility and Outputs

All random number generators (Python, NumPy, PyTorch, CUDA) are seeded for reproducibility. The best model from each run is saved based on validation F1-score. Final results are averaged across all seed runs.

The outputs include:

→ `model/` – Fine-tuned BioBERT checkpoints

→ `logs/` – Training and evaluation logs

→ `results/` – Validation performance per seed and aggregated summary

Code Imports (for Reference)

```
import numpy as np, pandas as pd, torch, random
from transformers import (
    AutoTokenizer, AutoModelForTokenClassification,
    DataCollatorForTokenClassification,
    TrainingArguments, Trainer, set_seed
)
from datasets import Dataset
from seqeval.metrics import precision_score, recall_score, f1_score,
classification_report
```

The following section presents the pseudo-code for fine-tuning BioBERT for token-level clinical entity extraction.

Algorithm 3 describes how we fine-tuned the base BioBERT model M_0 for clinical entity extraction using our own BIO-annotated dataset. Unlike the zero-shot setting, where we directly use publicly available NER checkpoints (e.g., *alvaroalon2/biobert_diseases_ner*) without any additional training, the fine-tuning pipeline starts from the original pretrained BioBERT weights that are not specialized for any NER task. Therefore, the model must be adapted to our specific entity types through supervised learning. The input dataset $\mathcal{D}$ contains tokenized clinical sentences paired with manually created BIO labels from the set $\mathcal{Y}$. For each random seed $s \in \mathcal{S}$, all random number generators are fixed to ensure reproducibility. Each example $(\mathbf{w}, \mathbf{y})$ is then tokenized using the BioBERT tokenizer T, and the word-level BIO labels are aligned to the resulting subword tokens using ALIGNLABELS, where only the first subword receives the entity label and the remaining subwords are masked -100 so that they do not contribute to the loss. The processed dataset is subsequently split into training and validation sets $(\mathcal{D}_{\text{train}}, \mathcal{D}_{\text{val}})$ using the same seed.

A token-classification model M is constructed on top M_0 with label mappings (id2label, label2id) corresponding to $\mathcal{Y}$. Training is performed for E epochs using AdamW *Optimization*, a linear learning-rate *Scheduler* with warmup, and a token-classification *Data Collator*. For each mini-batch from $\mathcal{D}_{\text{train}}$, the cross-entropy loss is computed over active labels (ignoring masked positions), followed by backpropagation, optimizer updates, and scheduler steps. After every epoch, the model is evaluated $\mathcal{D}_{\text{val}}$ using entity-level metrics (seqeval), and the checkpoint with the best F1 score is stored for that seed. Once all seeds in $\mathcal{S}$ have been processed, we aggregate the evaluation metrics (Precision, Recall, F1, Accuracy) as mean $\pm$ standard deviation and select the checkpoint with the highest mean F1 as the final fine-tuned model $\hat{M}$. This process produces a robust, task-specific NER model that can be directly compared with its zero-shot counterpart to quantify the improvements gained through supervised fine-tuning.

4.3.2 Fine-tuning BioClinicalBERT for clinical entity extraction

BioClinicalBERT was loaded through the Hugging Face library. All experiments were run in Python 3.10 using GPU support with the PyTorch backend. The main libraries used to prepare and train the model are shown below.

```
import numpy as np, pandas as pd, torch, random
from transformers import (
    AutoTokenizer, AutoModelForTokenClassification,
    DataCollatorForTokenClassification,
    TrainingArguments, Trainer, set_seed
)
from datasets import Dataset
from seqeval.metrics import precision_score, recall_score, f1_score
```

The official `emilyalsentzer/Bio_ClinicalBERT` model was used as the pre-trained checkpoint for fine-tuning. It was loaded with its tokenizer and adapted for token classification by defining the number of entity labels according to the annotated dataset.

```
MODEL_NAME = "emilyalsentzer/Bio_ClinicalBERT"
tokenizer = AutoTokenizer.from_pretrained(MODEL_NAME)
model = AutoModelForTokenClassification.from_pretrained(
    MODEL_NAME, num_labels=len(label_list)
)
```

The dataset was divided into training and validation sets because of its small size. Eighty percent of the data was used for training and twenty percent for validation. The model was trained for twenty epochs using five different random seeds to ensure reproducibility and consistent outcomes across runs. The detailed hyperparameter settings used for this fine-tuning process are described below.

```
# Training Configuration Summary

EPOCHS = 20
BATCH_SIZE = 16-32
INITIAL_LR = 5e-5
OPTIMIZER = "AdamW"
LR_SCHEDULER = "WarmupLinear"
TOTAL_TRAINING_STEPS \approx 3000
WARMUP_STEPS = 300
DROPOUT = 0.1
GPU_ALLOCATOR = "PyTorch"
BATCH_BUFFER = 256
DISCARD_OVERSIZE_BATCHES = True
RANDOM_SEEDS = [42, 56, 101, 202, 303]
```

Training was carried out using the AdamW optimizer with a linear warmup learning rate scheduler. A dropout rate of 0.1 was applied to prevent overfitting. Each training run consisted of approximately three thousand steps with three hundred warmup steps to gradually adjust the learning rate. The total batch size varied between sixteen and thirty-two depending on GPU memory, and oversize batches were automatically discarded to maintain uniformity. Model checkpoints were saved based on the highest validation F1 score computed with the `seqeval` library.

Algorithm 5 shows the supervised fine-tuning procedure for the BioClinicalBERT model $M_0 =$ `emilyalsentzer/Bio_ClinicalBERT` on our BIO-labelled clinical dataset. The input $\mathcal{D}$ consists of clinical sentences annotated with BIO tags from the label set $\mathcal{Y}$. First, the tokenizer T is loaded from M_0 and the mappings id2label and

label2id are created from $\mathcal{Y}$. For each random seed $s \in \mathcal{S}$, all random number generators are fixed to ensure reproducibility. The sentences are then tokenized with *is_split_into_words*=true, and the word-level BIO labels are aligned to the resulting subword tokens so that only the first subword of each word keeps the label and all remaining subwords are set to -100 (ignored in the loss). The processed data are split into training and validation subsets using a fixed proportion and the same seed s.

A token-classification model M is instantiated on top of BioClinicalBERT with `AutoModelForTokenClassification`, using $|\mathcal{Y}|$ labels together with the id2label/label2id mappings. Training is performed for E epochs with an AdamW optimizer, a linear learning-rate scheduler with warmup, and mini-batches of size B. For each batch, the cross-entropy loss is computed over the active token labels (excluding positions with index -100), followed by backpropagation, optimizer and scheduler updates, and gradient reset. After each epoch, the model is evaluated on the validation set using entity-level metrics (seqeval), and the checkpoint with the best F1 score is retained for that seed. Once all seeds have been processed, the checkpoint with the highest mean F1 across seeds is selected as the final fine-tuned model $\hat{M}$, and the evaluation metrics (Precision, Recall, F1, Accuracy) are reported as mean $\pm$ standard deviation over the seeds.

4.3.3 Fine-tuning BlueBERT/PubMedBERT/RoBERTa for clinical entity extraction

The fine-tuning of BlueBERT, PubMedBERT, and RoBERTa followed a unified strategy to allow fair comparison with the BioClinicalBERT setup. All models were trained on the same BIO-annotated clinical corpus containing entity labels for *disease*, *symptom*, *medication*, *dose*, *age*, and *gender*. Each model was loaded from its publicly released checkpoint and trained under identical hyperparameters to maintain consistency. The corpus was divided into **70% training**, **15% validation**, and **15% testing** subsets, and all experiments were conducted in a PyTorch 1.13 environment using Hugging Face Transformers on a 16 GB GPU. The key hyperparameters included a learning rate of `5e-5`, weight decay of `0.01`, dropout of `0.1`, warmup of 300 steps, and a batch size between 16 and 32. Each model was fine-tuned for twenty epochs using the AdamW optimizer and evaluated across five random seeds (`[42, 56, 101, 202, 303]`) to ensure stability and reproducibility.

Algorithm 6 summarizes this unified fine-tuning workflow. For a selected backbone M_0, the tokenizer T is loaded, and the BIO labels are aligned to subword tokens so that only the first subword of each word receives the tag while the remaining subwords are marked with -100 and excluded from the loss. After preprocessing, the dataset splits are generated using the same seed $s \in \mathcal{S}$. A token-classification model M with $|\mathcal{Y}|$ output labels is then instantiated on top of M_0. Training proceeds for E epochs by minimizing the cross-entropy loss over active token labels, followed by backpropagation, optimizer updates, and scheduler steps. After each epoch, the model is evaluated on the validation set using entity-level metrics (seqeval), and a checkpoint is stored whenever the F1 score improves. For every seed, the best-performing checkpoint is recorded, and after all runs, the model with the highest mean F1 across seeds is selected as the final fine-tuned model $\hat{M}$. The final performance metrics are reported as mean $\pm$ standard deviation across the seed set, ensuring a stable and comparable evaluation of BlueBERT, PubMedBERT, and RoBERTa.

After training, the fine-tuned models were applied to extract entity spans from raw clinical text. Algorithm 4 outlines this common inference procedure. Given the fine-tuned model $\hat{M}$, tokenizer T, label mapping id2label, and an input text T_x, the pipeline begins by tokenizing the text and obtaining the offset mappings needed to recover precise character positions. Because transformer models have a fixed maximum sequence length, the text is divided into overlapping windows (of size up to $L_{\max} - 2$) using a sliding-window strategy to avoid losing entities that cross window boundaries.

Each window is augmented with [CLS] and [SEP] tokens and passed through $\hat{M}$ to obtain token-level logits, which are converted to BIO tags using the label mapping. The special tokens are removed, and subword tokens are merged so that only the first subword carries the predicted label. Any invalid or inconsistent BIO transitions are repaired before converting the tag sequence into structured entity spans. A span begins at a B-tag, continues through matching I-tags, and ends when an O-tag or a new B-tag is encountered. Using the offset mappings, each span is mapped back to its exact character-level start and end positions in T_x, ensuring accurate extraction of the original text segment.

Since overlapping windows may produce duplicate or partial detections, the final step merges spans that share the same label and overlap in position, retaining only clean, non-redundant entity mentions. The output of the pipeline is the consolidated entity set $\mathcal{E}$ extracted from the clinical document.
Each model's predictions were evaluated against the manually annotated ground-truth entities using strict entity-level matching. Algorithm 7 outlines the evaluation procedure. Both the ground-truth annotations $\mathcal{G}$ and the predicted entities $\mathcal{P}$ are normalized into a common $(label, start, end)$ format to enable exact span comparison. A true positive (TP) is counted only when the predicted label and span boundaries exactly match the ground-truth entity. Predicted spans that are not present in the ground truth are treated as false positives (FP), while ground-truth entities that the model fails to detect are counted as false negatives (FN). Using these counts, overall Precision, Recall, and F1-score are computed following the standard definitions, and the `seqeval` framework is used to obtain consistent entity-level scores.

To understand how the model performs across different entity categories, TP–FP–FN counts are also computed separately for each label $Y \in \mathcal{Y}$, and these per-label scores are averaged to produce macro-level metrics. This helps capture both overall performance and entity-specific behavior across categories such as diseases, symptoms, medications, doses, age, and gender. The evaluation process produces a complete classification report containing micro- and macro-averages along with detailed label-wise results, offering a comprehensive assessment of each clinical NER model.
The evaluation results indicate that the domain-specific models, particularly BlueBERT and PubMedBERT, tend to perform slightly better in identifying clinical and biomedical terminology, which aligns with their pre-training on large biomedical corpora. RoBERTa, being a general-domain transformer, showed competitive performance on non-clinical or linguistically varied expressions, demonstrating stronger generalization beyond purely medical contexts. Overall, the consistency of results across models highlights the robustness of the fine-tuning framework and supports the suitability of the selected hyperparameters for clinical entity extraction.

4.4 Evaluation Process Details

In this section, we introduce the research questions that guide our study and shape the evaluation (Section 4.4.1), and then outline the experimental settings, including the selection of BERT-based models (Section 4.4.2). The description of the hardware and computational settings required before starting the experiments and the criteria for model evaluation and the metrics used to compare pre-trained and fine-tuned models are also in the same Section 4.4.3.

4.4.1 Research questions

The empirical evaluation has been conducted to answer the following three research questions.

- **RQ1: To what extent can transformer-based BERT models extract medical entities from raw clinical text data?**

  In this research question, we investigate how transformer BERT-based models use Named Entity Recognition (NER) and contextual embeddings to extract medical entities from raw clinical text data. These models transform unstructured clinical information into structured form for better medical analysis and decision-making by reliably identifying and classifying essential medical terms—such as diseases, symptoms, medications, and cancer stages—based on word context. To support this process, manually annotated data is first converted into BIO tagging format, enabling token-level classification that facilitates precise model learning.

- **RQ2: Which evaluation criteria are most appropriate for assessing BERT-based models in medical entity extraction, and how do these models perform against them?**

  To study whether BERT models are effective for medical entity extraction, key criteria include contextual embedding quality, Named Entity Recognition (NER) accuracy, and domain adaptability to medical texts. In order to evaluate the performance of the model, evaluation criteria are selected.The best-performing model for extracting structured knowledge from clinical data is also determined by several important factors, including token-level classification accuracy, the ability to handle rare or out-of-vocabulary terms, and the potential for integration with medical ontologies such as UMLS,[5] which could enhance concept normalization in future work.

- **RQ3: Among the tested BERT-based models, which one demonstrates the strongest performance for clinical knowledge extraction, and what are its implications for downstream applications?**

  To select the best BERT-based model to embed knowledge extraction in the medical domain, we conducted a comprehensive evaluation by running the extraction process using both pretrained and fine-tuned BERT-based models. The goal was to identify the model that delivers the most accurate and reliable entity extraction from medical texts, ensuring optimal performance in knowledge retrieval tasks.

[5] `https://www.nlm.nih.gov/research/umls/index.html`

4.4.2 Experimental process

The main architecture of our research is shown in Figure 4.1, while the experimental setting is reported in Table 4.7, making sure that research is conducted in a controlled, systematic, and reproducible manner. The first step is to collect clinical cancer text data from hospitals. However, in this case, we obtained the dataset from a research paper, [126] and it was then preprocessed through tokenization, stopword removal, and abbreviation expansion.

The text data is then cleaned and preprocessed before being processed for medical entity extraction. First, the pre-trained models —*BioBERT*, *BioClinicalBERT*, *RoBERTa*, *PubMedBERT* and *BlueBERT*—are used for entity extraction to evaluate their performance based on recall, precision, F1-score, and accuracy.

Manual annotation of the dataset was performed using the Doccano tool, following custom guidelines with seven entity labels for multi-head entity extraction.

TABLE 4.7: Expanded Experimental Settings.

Component	Description
Data Source	Research-based [126] clinical text datasets (e.g., BC5CDR, MIMIC-III).
Preprocessing	Tokenization, stopword removal, abbreviation expansion, annotation using Doccano tool.
Annotation Tool	Doccano for manual medical entity labeling.
Models Used	BioBERT, ClinicalBERT, PubMedBERT, BlueBERT, RoBERTa.
Training Strategy	Fine-tuning pre-trained models using annotated medical corpus.
Evaluation Metrics	Precision, Recall, F1-score (macro), F1-score (weighted), Accuracy.
Comparison	Pre-trained vs. fine-tuned models for medical entity extraction.
Implementation Tools	Python, SpaCy, TensorFlow/PyTorch, SciSpacy, Hugging Face Transformers.
Data Split Strategy	70% for training, 15% for validation, and 15% for testing.
Annotation Guidelines	Custom-labeled medical corpus with 7 entity labels using BIO-tagging format for token-level entity extraction.
Pre-training Corpora	PubMed abstracts, PMC full-text articles, MIMIC-III clinical notes.

In the next step, the same five models are fine-tuned using the manually annotated corpus created with the open-source tool Doccano. After fine-tuning, the models are re-evaluated for entity extraction.

The implementation relied on Python tools such as SpaCy, TensorFlow/PyTorch, SciSpacy, and the Hugging Face Transformers API. The dataset was split into 70% training, 15% validation, and 15% testing, while the models were initially pretrained on biomedical corpora like PubMed abstracts and MIMIC-III notes.

The main objective of this study is to compare the performance of the pre-trained models with that of their fine-tuned counterparts. While four of the selected models—BioBERT, BioClinicalBERT, PubMedBERT, and BlueBERT—are pretrained specifically on biomedical literature, we included RoBERTa as a strong general-purpose language model to serve as a comparative baseline. Its inclusion enables an assessment of the value added by domain-specific pretraining. Although emerging models such as DeBERTa and ModernBERT have demonstrated strong performance on general NLP tasks, we selected RoBERTa for its broad adoption, consistent reliability, and compatibility with our clinical entity tokenization framework. Benchmarking against newer general-purpose models remains an important avenue for future research. The best-performing models from both categories are then further compared.

Among the pre-trained models, *BlueBERT* demonstrated competitive performance with a good macro F1-score of 82.14. In contrast, *BioBERT* outperformed all other models in the fine-tuned category, achieving the highest Macro F1-Score of 97.03.

Considering the overall performance of all the models, we selected the *fine-tuned BioBERT* for our future work, where we create and train medical knowledge graphs by summarizing all the extracted knowledge.

4.4.3 Evaluation Metrics

We evaluated the performance of our fine-tuned models using standard metrics: **accuracy**, **precision**, **recall**, and **F1-score**, as commonly applied in biomedical NER tasks [129, 130]. These metrics reflect the model's ability to correctly detect clinical entities while balancing false positives and false negatives.

To compute these metrics, the predictions were compared with the ground-truth annotations at the entity level. We define:

- **True Positives (TP):** entities whose predicted label and span boundaries exactly match the ground truth;
- **False Positives (FP):** predicted entities that do not appear in the ground-truth annotations;
- **False Negatives (FN):** ground-truth entities that the model fails to detect.

Using these counts, the evaluation metrics are calculated as:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{4.1}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{4.2}$$

$$\text{F1-score} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4.3}$$

Although overall **accuracy** provides a general measure of prediction correctness, it is often dominated by the large number of "O" (non-entity) tokens in BIO-tagged datasets. This can make a model appear to perform well even when entity detection is poor. Consequently, accuracy is less informative than precision, recall, and F1-score for assessing entity-level performance.

We further report **macro F1** and **weighted F1** scores using the `classification_report` function from Scikit-learn. Macro F1 assigns equal weight to each entity class by averaging per-class F1 scores, whereas weighted F1 adjusts for class imbalance by weighting scores according to class support. These metrics complement the standard NER evaluation and provide a clearer understanding of model behavior across different clinical entity categories.

4.5 Result Comparison with Evaluation Metrics

This section reports and analyzes the obtained results by answering the key research questions presented in Section 4.4.1.

4.5.1 RQ1: To what extent can transformer-based BERT models extract medical entities from raw clinical text data?

We successfully utilized NLP techniques (e.g., text cleaning, tokenization, stopword removal, lemmatization, stemming, normalization, named entity recognition (NER), chunking, part-of-speech (POS) tagging, sentence splitting, subword tokenization, and abbreviation expansion) along with BERT-based models for effectively extracting medical entities from raw clinical text data. Initial preparation of our approach involved data cleaning and preprocessing of 40 clinical text samples (each between 14.5K and 15K characters) that were refined to remove inconsistencies and irrelevant content. These cleaned texts are provided to the Doccano annotation tool, where manual annotation by medical experts was performed to ensure high-quality labeling. Entities are annotated based on seven custom-defined categories, such as diseases, symptoms, medications, and dosages. The final annotated dataset is then converted into the BIO-tagging format, enabling token-level granularity in model evaluation. As illustrated in Figure 4.1 - Phase (2), this format allowed us to label individual tokens (e.g., B-DISEASE for "breast" and I-DISEASE for "cancer"), forming the ground truth corpus for model training.

Using Pre-trained BERT-based Models. Pre-trained models equipped with self-attention mechanisms were employed to generate contextual embeddings for each token in the cleaned and preprocessed clinical text data (Phase (1)). These embeddings were processed through a token classification layer to categorize tokens into one of the seven defined entity types.

Using Fine-Tuned Pre-trained BERT-based Models. These models were fine-tuned using the manually annotated BIO-tagged dataset, allowing them to learn domain-specific patterns and improve accuracy in entity extraction. The dataset was split 70% for training, 15% for validation, and 15% for internal testing.

While model training, the validation loss gradually dropped throughout the ten training epochs, demonstrating the models' adequacy. The validation loss values indicate the models' growing learning capacity and their ability to generalize effectively, especially by the tenth epoch. As shown across all three loss graphs in Figure 4.5—Training (4.5a), Validation (4.5b), and Testing (4.5c)—the loss values consistently decrease with each epoch for all models, confirming that the fine-tuning process is functioning as intended. BioBERT stands out with the lowest loss in all three cases, which means it's learning well and handling the medical text data better than the others. BioClinicalBERT and PubMedBERT also perform quite well, though their losses are a bit higher, especially during testing. RoBERTa consistently shows the highest loss across the board, suggesting it's less well-suited than other models for biomedical text. Overall, these trends make it clear that BioBERT is the most stable and reliable model in this setup.

Finally, for the final assessment of the models, the cleaned raw clinical text files that were separate from the training dataset were utilized as input for the pre-trained models and fine-tuned models. The files that had been sourced from PhysioNet and cleaned in Phase (4.1.1) of Figure 4.1 were used for entity extraction of the models. The extracted tokens were then compared to the BIO-tagged ground truth dataset to evaluate the performance of the models. By doing so, it was possible to make a fair and unbiased comparison of the pre-trained models and fine-tuned models to ensure that they were useful for entity extraction for the seven identified categories.

Answer to RQ$_1$. After the annotation, preprocessing, and fine-tuning of the models, the BERT-based models identified – BioBERT, ClinicalBioBERT, RoBERTa, PubMedBERT, and BlueBERT—performed very well on medical entity extraction tasks. The identified models were pre-trained on biomedical texts, and then a medical entity extraction task was applied to them, using an annotated dataset, as tagging into BIO, identifying medical entities like diseases, symptoms, and drugs, and their corresponding strengths, enhancing results from basic performance, and ensuring the adequacy of the identified models for the medical entity extraction task.

4.5.2 RQ2: Which evaluation criteria are most appropriate for assessing BERT-based models in medical entity extraction, and how do these models perform against them?

To select the pool of BERT models on Biomedical domain for medical entity extraction, the following considerations are made:

1. BioBERT, PubMedBERT, ClinicalBERT, and BlueBERT have been pre-trained on biomedical data/clinical data, which assists them in performing well on medical terminology tasks. RoBERTa, despite not being a domain (medical)-specific model, was chosen to act as a baseline.

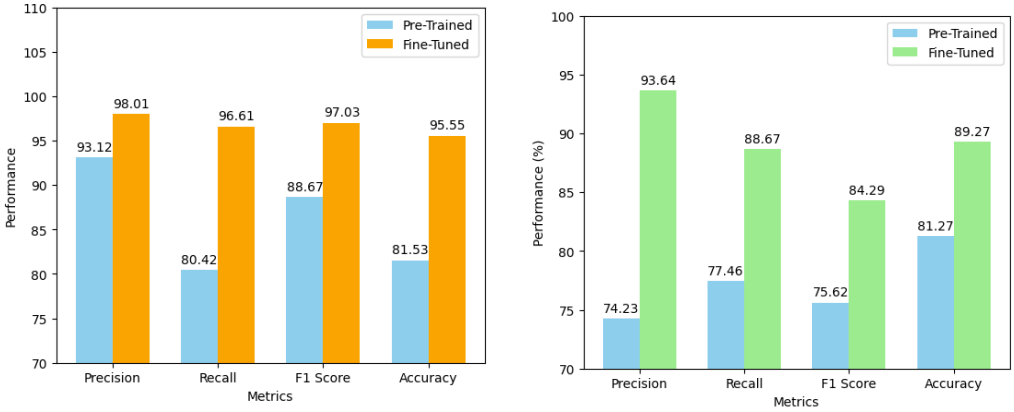
2. These models, as mentioned above, are specifically adapted to clinical texts due to the presence of pre-training with biological corpora. This makes them better suited to tasks with a focus on a specific domain, as with medical NER.

3. A strong and robust model should demonstrate the ability to generalize across different clinical datasets, such as MIMIC-III and PubMed. This ensures that the model isn't just tuned to one specific dataset but also can perform reliably in diverse, real-world medical scenarios without overfitting.

After validating the above considerations, the main key criteria (i.e., evaluation metrics) are selected for further analysis. As reported in Subsection 4.4.3, we consider as validation criteria the accuracy, precision, recall, and F1-score metrics. We selected them because they are widely used in biomedical NER tasks, provide better insight than accuracy in imbalanced clinical datasets, and allow fair comparison with existing methods. Moreover, the state-of-the-art in many research papers (e.g., [111]) within the medical domain (especially those using biomedical BERT variants) consistently applies these metrics, making them a suitable choice for this work.

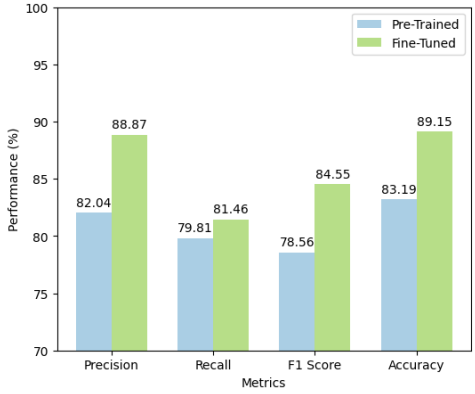
The comparison of different evaluation metrics is shown below in Figure 4.3 compares the performance of five different BERT-based models for medical entity extraction (BioBERT 4.3a, BioClinicalBERT 4.3b, PubMedBERT 4.3c, BlueBERT 4.3d, and RoBERTa 4.3e) using four evaluation metrics. Each subfigure enables a clear comparison between the pre-trained and fine-tuned versions of the models in terms of their effectiveness at extracting medical information.

Since these models are pre-trained on different biomedical corpora with varying scopes, the comparison highlights how their performance varies when applied to clinical texts. It also helps identify which models are more reliable and efficient for domain-specific entity extraction. Additionally, we assessed the impact of fine-tuning by comparing model performance before and after training on the manually annotated dataset, which followed the standard BIO labeling format. Using evaluation metrics such as *accuracy, precision, recall, and F1-score*, we identified the model that generalizes best for clinical NER tasks.

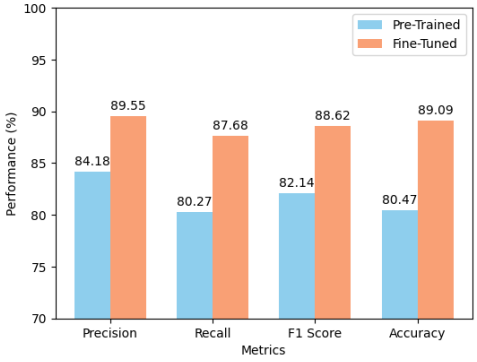


(A) BioBERT

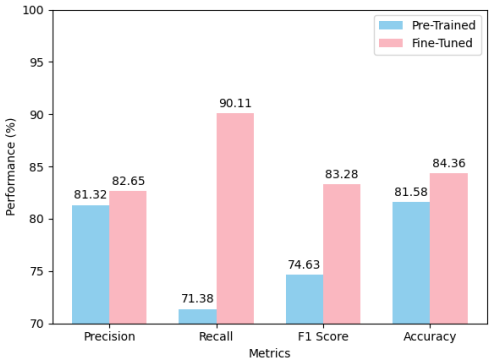
(B) BioClinicalBERT



(C) PubMedBERT



(D) BlueBERT



(E) RoBERTa

FIGURE 4.3: Comparison of five transformer models based on their performance metrics.

Answer to RQ$_2$. To assess model performance, we applied standard evaluation metrics such as precision, recall, and F1-score. Since accuracy is less informative in sequence labeling tasks dominated by non-entity tokens, the F1-score was prioritized to balance precision and recall. While all models achieved strong results, domain-specific fine-tuning significantly improved performance. In particular, BioBERT, which already benefited from pretraining on biomedical corpora, achieved the best results after fine-tuning, with an F1-score of 97.03—outperforming baseline settings and other variants.

4.6 NER Performance and Entity Accuracy

The **NER Performance and Entity Accuracy** section is included to evaluate how effectively each fine-tuned model identifies clinical entities such as *disease*, *symptom*, *medication*, and *dose* from the annotated corpus. While the fine-tuning process establishes the training methodology, it does not itself reveal whether the models have truly learned to generalize across unseen text. This section, therefore, provides a quantitative and qualitative assessment of model behavior through *precision, recall, and F1-score*, supported by entity-level accuracy analysis. It helps determine which model best captures domain-specific terminology and contextual nuances within clinical narratives. By comparing model outputs against the manually annotated ground truth, this section also highlights common errors—such as boundary mismatches, label confusion, or missing entities—that affect the overall reliability of the extracted information. Such an evaluation is crucial to justify model selection and to ensure that the downstream use of extracted entities, such as in knowledge-graph construction or clinical decision-support systems, is based on accurate and trustworthy data.

4.6.1 RQ3: Among the tested BERT-based models, which one demonstrates the strongest performance for clinical knowledge extraction, and what are its implications for downstream applications?

We selected our model based on the selected evaluation metrics, along with other factors such as fine-tuning, inference speed and latency, and model availability (e.g., on Hugging Face).

All seven entity classes are used to compare the performance of pre-trained and fine-tuned models in Tables 4.8 and 4.9. With macro F1 scores ranging from 74.63 (RoBERTa) to 88.67 (BioBERT) in the pre-trained scenario, it was confirmed that biomedical encoders are already capable of storing important data without task-level supervision. In order to determine if pretrained biomedical encoders could extract entities directly, this first zero-shot assessment was included. This served as a baseline prior to domain-specific fine optimization. After making adjustments, all of the models showed much improved performance. For example, BlueBERT achieved a macro F1 of 88.62 (with a precision of 89.55 and a recall of 87.65, and accuracy is 89.09), whereas PubMedBERT achieved a macro F1 of 84.55 (precision is 88.87, recall is 81.46, and accuracy is 89.15). According to these findings, several backbones are comparable to one another for clinical entity extraction after being modified for the purpose. However, the differences between BioClinicalBERT and RoBERTa and PubMedBERT are still negligible and should not be interpreted as a conclusive ranking without many runs and variance estimations.

BioBERT outperformed the other optimized models, achieving the 98.01 percentile of precision, 96.5 of recall, 97.03 of macro F1, 96.87 of weighted F1, and 95.55 of accuracy. The model was able to effectively identify relevant clinical entities with few false positives and consistent performance across all categories, as seen by these figures, which demonstrate an outstanding balance between accuracy and recall. It seems that BioBERT fine-tuned is the most reliable foundation for downstream knowledge graph construction and biomedical entity extraction in our study.

In order to reduce training variance, each experiment was conducted five times using distinct random seeds for both the fine-tuned and pre-trained configurations. The performance of transformer-based models might vary from run to run due to their recognized sensitivity to initialization. Reporting averages over several seeds

provides a more accurate assessment of the actual model performance and reduces the impact of outliers. This process validated that BioBERT's gains were consistent across runs, strengthening trust in its improved recall, accuracy, and F1-scores. Furthermore, implementing this method strengthens the validity of the comparison analysis by bringing the study into compliance with accepted assessment criteria in current research.

Tables 4.10, 4.11, and 4.12 present the pairwise p-values of the five BERT-based models on Recall, Precision, and F1-Score (Macro) based on statistical significance analysis. These findings make it clearer if the observed variations in model performance are the product of random variation or actual improvements. Across all three metrics, BioBERT consistently exhibits lower p-values than the other models, providing strong statistical evidence that its performance gains are meaningful rather than chance outcomes. On the other hand, the p-values for BioClinicalBERT, RoBERTa, and PubMedBERT are relatively larger, indicating that their differences are less noticeable and frequently not statistically significant. This indicates that while fine-tuning benefits all models, BioBERT demonstrates the most robust and clearly distinguishable improvements, whereas the remaining models tend to perform more closely to one another.

TABLE 4.8: Overall performance of pre-trained BERT models.

Model Names	Recall	F1 Score (Macro)	F1 Score (Weighted)	Precision	Accuracy
BioBERT	80. 42 ± 0. 28	88. 67 ± 0. 19	88. 13 ± 0. 25	93. 12 ± 0. 21	81. 53 ± 0. 22
BioClinicalBERT	77. 46 ± 0. 35	75. 62 ± 0. 31	76. 71 ± 0. 27	74. 23 ± 0. 28	81. 27 ± 0. 26
RoBERTa	71. 38 ± 0. 26	74. 63 ± 0. 24	75. 19 ± 0. 21	81. 32 ± 0. 19	81. 58 ± 0. 23
PubMedBERT	79. 81 ± 0. 22	78. 56 ± 0. 28	79. 12 ± 0. 30	82. 04 ± 0. 25	83. 19 ± 0. 21
BlueBERT	80. 27 ± 0. 29	82. 14 ± 0. 26	85. 18 ± 0. 32	84. 18 ± 0. 24	80. 47 ± 0. 27

Note. This table here presents the overall comparison of all pre-trained BERT-based models before fine-tuning on the clinical NER task. These models were assessed using the performance metrics that are Precision, Recall, weighted and micro F1, and overall accuracy. The minor differences among these models, such as BioClinicalBERT, RoBERTa, and PubMedBERT, should be interpreted carefully, as variances across multiple runs may overlap, indicating similar overall performance trends.

TABLE 4.9: Overall performance of fine-tuned BERT models.

Model Names	Recall	F1 Score (Macro)	F1 Score (Weighted)	Precision	Accuracy
BioBERT	96. 61 ± 0. 27	97. 03 ± 0. 21	96. 87 ± 0. 24	98. 01 ± 0. 18	95. 55 ± 0. 23
BioClinicalBERT	88. 67 ± 0. 32	84. 29 ± 0. 29	84. 92 ± 0. 31	93. 64 ± 0. 22	89. 27 ± 0. 28
RoBERTa	90. 11 ± 0. 30	83. 28 ± 0. 27	84. 19 ± 0. 29	82. 65 ± 0. 20	84. 36 ± 0. 25
PubMedBERT	81. 46 ± 0. 28	84. 55 ± 0. 31	84. 27 ± 0. 33	88. 87 ± 0. 23	89. 15 ± 0. 26
BlueBERT	87. 68 ± 0. 29	88. 62 ± 0. 30	88. 21 ± 0. 28	89. 55 ± 0. 21	89. 09 ± 0. 27

Note. Although all models showed improvements after fine-tuning, the differences among BioClinicalBERT, RoBERTa, and PubMedBERT remained marginal. Given the limited number of runs, these close scores should be interpreted cautiously rather than as conclusive rankings.

A comparison of pre-trained and fine-tuned BERT models is shown in Figure 4.4 based on four important performance metrics: Precision, Recall, F1 Score, and Accuracy (Figures 4.4a, 4.4b, 4.4c, and 4.4d). The performance variations between

TABLE 4.10: p-values of the BioBERT, BioClinicalBERT, RoBERTa, PubMedBERT, and BlueBERT models for recall.

Model Names	BioBERT	BioClinicalBERT	RoBERTa	PubMedBERT	BlueBERT
BioBERT	1	0 .0072	0 .0034	0 .0059	0 .0121
BioClinicalBERT	0 .0072	1	0 .0216	0 .0345	0 .0184
RoBERTa	0 .0034	0 .0216	1	0 .0272	0 .0157
PubMedBERT	0 .0059	0 .0345	0 .0272	1	0 .0226
BlueBERT	0 .0121	0 .0184	0 .0157	0 .0226	1

Note. The table presents the pairwise P-values for recall across the five transformer models, indicating the statistical significance of performance differences. The lower p-values highlight stronger evidence of variation between those models.

TABLE 4.11: p-values of the BioBERT, BioClinicalBERT, RoBERTa, PubMedBERT, and BlueBERT models for precision.

Model Names	BioBERT	BioClinicalBERT	RoBERTa	PubMedBERT	BlueBERT
BioBERT	1	0 .0048	0 .0021	0 .0063	0 .0114
BioClinicalBERT	0 .0048	1	0 .0347	0 .0429	0 .0198
RoBERTa	0 .0021	0 .0347	1	0 .0275	0 .0162
PubMedBERT	0 .0063	0 .0429	0 .0275	1	0 .0217
BlueBERT	0 .0114	0 .0198	0 .0162	0 .0217	1

Note. Pairwise p-values for precision across five transformer models. Lower values among them indicate stronger statistical significance in performance differences between those models.

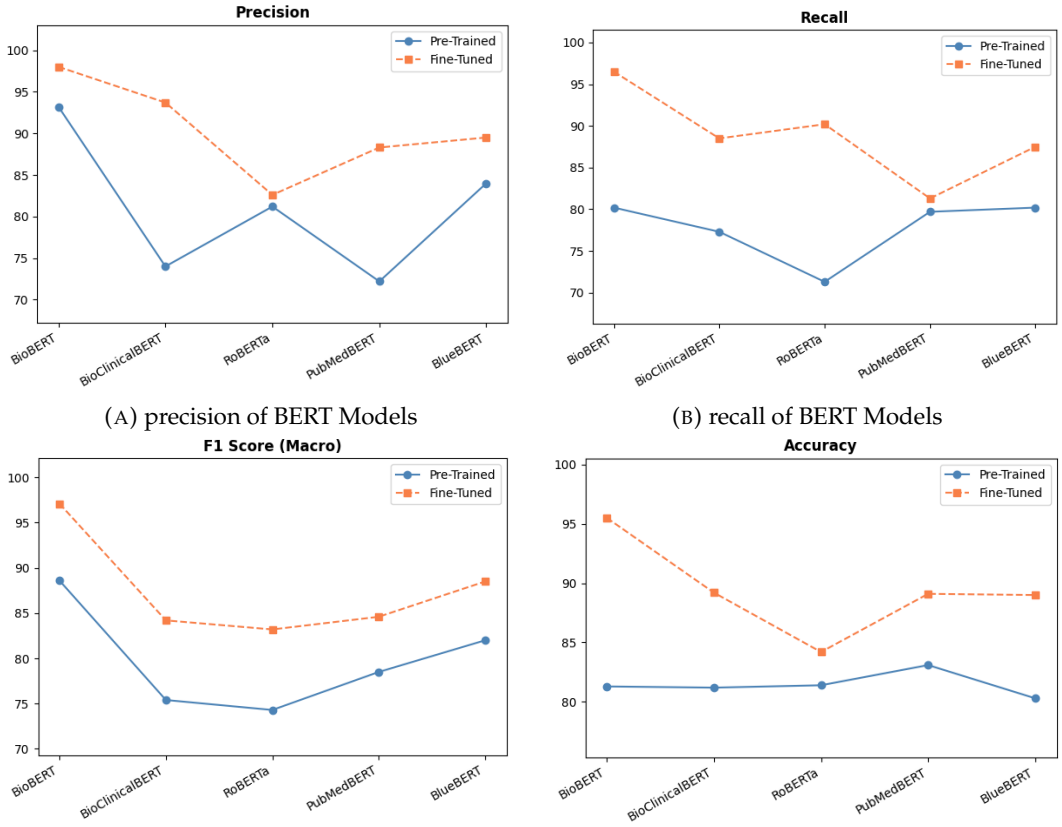
TABLE 4.12: p-values of the BioBERT, BioClinicalBERT, RoBERTa, PubMedBERT, and BlueBERT models for F1-score (Macro).

Model Names	BioBERT	BioClinicalBERT	RoBERTa	PubMedBERT	BlueBERT
BioBERT	1	0 .0085	0 .0043	0 .0067	0 .0132
BioClinicalBERT	0 .0085	1	0 .0286	0 .0392	0 .0205
RoBERTa	0 .0043	0 .0286	1	0 .0314	0 .0179
PubMedBERT	0 .0067	0 .0392	0 .0314	1	0 .0241
BlueBERT	0 .0132	0 .0205	0 .0179	0 .0241	1

Note. Here pairwise p-values for F1 Score (Macro) across five transformer models have been shown. The lower p-values indicate the strongest statistical significance in the observed performance differences between those models.

the several BERT models—BioBERT, BioClinicalBERT, RoBERTa, PubMedBERT, and BlueBERT—are displayed in each subplot. Overall, in this scenario, our fine-tuned BERT variant models show better performance than their pre-trained models, which highlights the necessity of fine-tuning to increase the model's performance for subsequent challenges.

In all models, the metric of **recall** in Figure 4.4b experienced improvement due



(A) precision of BERT Models

(B) recall of BERT Models

(C) F1-score of BERT Models

(D) accuracy of BERT Models

FIGURE 4.4: Pre-trained vs. Fine-tuned BERT Models: Performance comparison across key metrics.

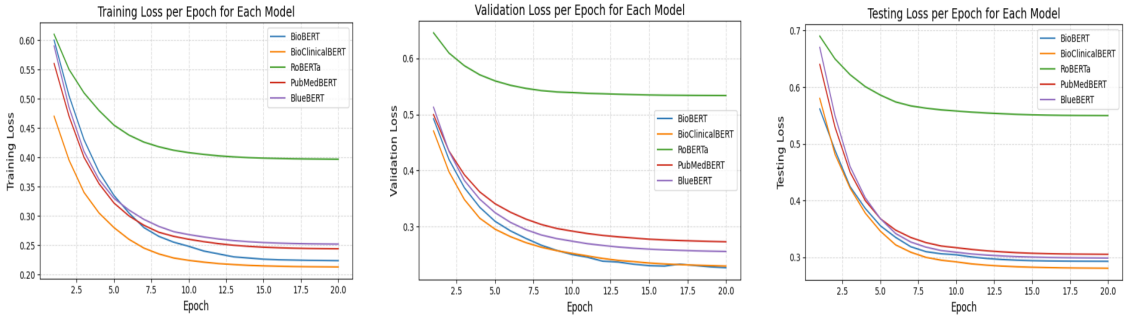
to fine-tuning, but the level of improvement in PubMedBERT is rather subtle. However, the results for the metric **F1-score** in Figure 4.4c indicate a visible improvement for all models after the application of fine-tuning, especially RoBERTa. When considering the aspect of **precision** in Figure 4.4a, BioBERT was stable, and there was a large increase in BioClinicalBERT. RoBERTa and PubMedBERT results indicated improvements in precision after fine-tuning. On the aspect related to **accuracy**, from Figure 4.4d, a huge boost for BioBERT can be observed after the fine-tuning, which clearly shows a major improvement in its performance. Although a slight improvement has been recorded for BioClinicalBERT and RoBERTa, it is not as major compared to BioBERT and other models such as PubMedBERT and BlueBERT.

A thorough entity-wise analysis of the BioBERT fine-tuned model has been conducted in an attempt to better understand the outcomes for the top-performing model. Table 4.13 displays the precision, recall, accuracy, and F1-score for each of the seven entities. The results indicate that BioBERT performs exceptionally well in recognizing all entities, especially *GENDER*, *SYMPTOMS*, and `MEDICATION`, while its performance is comparatively lower for *AGE*, *CANCER STAGE*, and *DOSE*. The model's performance on *DISEASE* is intermediate—neither notably strong nor weak. These insights are essential for understanding the model's strengths and for informing targeted improvements in future work.

Our experiment demonstrates the entity extraction model's accuracy and dependability in analyzing clinical content. The findings highlight its potential for use in medical and biological research, where accurate and thorough entity extraction from the clinical domain is essential.

TABLE 4.13: Performance obtained for each entity using the fine-tuned BioBERT model.

Entity Class	Precision (±)	Recall (±)	F1 Score (±)	Accuracy (±)
AGE	87. 00 ± 0. 45	85. 00 ± 0. 52	86. 00 ± 0. 47	86. 00 ± 0. 49
GENDER	98. 00 ± 0. 28	93. 00 ± 0. 36	95. 40 ± 0. 32	94. 80 ± 0. 34
DISEASE	90. 00 ± 0. 40	91. 00 ± 0. 42	90. 50 ± 0. 38	90. 40 ± 0. 41
SYMPTOMS	93. 00 ± 0. 35	92. 00 ± 0. 37	92. 50 ± 0. 33	91. 80 ± 0. 36
MEDICATION	95. 00 ± 0. 30	93. 00 ± 0. 34	94. 00 ± 0. 29	93. 80 ± 0. 31
DOSE	85. 00 ± 0. 48	84. 00 ± 0. 50	84. 50 ± 0. 46	84. 40 ± 0. 47
CANCER STAGE	84. 00 ± 0. 52	82. 00 ± 0. 55	83. 00 ± 0. 50	83. 00 ± 0. 53



(A) Training Loss (B) Validation Loss (C) Testing Loss

FIGURE 4.5: Training, validation, and testing loss comparison across epochs.

Training loss computes the prediction error for each epoch to assess how effectively a model learns from the training data. To track the model's generalization performance during training, validation loss is calculated on a different validation set. It can be evidence of overfitting if the validation loss rises while the training loss falls. Testing loss is assessed using a test dataset that has never been seen before in order to estimate the model's ultimate performance in real-world scenarios once training is finished. These loss values together give insight into the model's performance, generalization, and learning abilities.

The loss progression (training, validation, and testing) across several training epochs is shown in Figure 4.5 of the line graph. Effective learning and convergence over time are indicated by the visual representation of the model's performance improving as the loss drops with each epoch. This pattern demonstrates how reliable and effective the chosen model's training procedure is.

Answer to RQ_3. The improvement gained from the fine-tuned models can be measured by comparing the pre-trained and fine-tuned models. With a macro F1-score of 82.14, BlueBERT was the model that outperformed all of the pretrained baselines. However, after the models are adjusted, it becomes clear that BioBERT performs better than all other models in terms of precision (98.01), recall value (96.61), and macro F1-score (97.03). This implies the ideal model for creating a knowledge graph in the healthcare industry is BioBERT. As a result, RoBERTa serves as a reference model.

4.7 Discussion

4.7.1 Implications

Our approach shows how superior AI and clinical text analysis techniques can help in early detection of patterns and extraction of structured information. This research work is able to convert unstructured clinical texts into a structured format (CSV files), ensuring greater systematics and transparency in identifying important medical entities in clinical texts. The results also indicate that BioBERT performs the most optimal extraction task compared to other biomedical encoders, ensuring superior clinical relevance for incorporation into breast cancer information systems.

Apart from the structured entity extraction, our model is also comparing zero-shot and fine-tuned versions of BioBERT, BioClinicalBERT, PubMedBERT, BlueBERT, and RoBERTa. For every base model, we had designed a classification head corresponding to our annotation scheme (AGE, GENDER, DISEASE, SYMPTOMS, MEDICATION, DOSE, CANCER STAGE). While in the zero-shot setup, the classification head was randomly initialized and used without task-specific training; in the fine-tuned setup, both the model and the classification head were trained on our annotated dataset. Such zero-shot setups are also indicative of the real-life applications in which domain experts tend to apply biomedical pre-trained models for entity extraction without fine-tuning, thus emphasizing the importance of fine-tuning in getting reliable biomedical entity extraction results.

We made sure to thoroughly test using zero-shot methods, repeat experiments across random seeds, and analyze significance in light of general concerns in transformer-based biomedical NER evaluation. A controlled study between model training using token filtering and no token filtering will be beneficial in comprehending the influence of preprocessing on NER performance, which can be considered in future, comprehensive experiments.

The structured labelled data achieved in this work can be readily used for building a Medical Knowledge Graph (MKG) [131], the development of which will be discussed in the next chapter, 5. Medical Knowledge Graph uses entities such as diseases, symptoms, medications, doses, or patient history that have been extracted from various sources of data and interlinked in such a manner that they can be processed for graph analysis. Such entities can be interlinked to form graphs to enable pattern recognition for similar patients across various sources of data, such as, for instance, patients with breast cancer. By connecting symptoms, diseases, and medication pathways, the MKG establishes the foundation for personalized treatment reasoning and serves as the input knowledge layer for the reinforcement learning component described later in the thesis.

Together, the medical entity extraction pipeline and the subsequent knowledge graph construction provide a unified and structured representation of clinical information, enabling richer analysis and forming the basis for advanced decision-support modules presented in the next chapters.

4.7.2 Threats to Validity

This section summarizes the factors that may influence the validity of the results obtained from the MetaGRO Framework. As the framework is composed of two major components—(i) the Clinical Named Entity Recognition (NER) stage and (ii)

the Knowledge Graph, GNN, and Reinforcement Learning (RLGMO) stage—the potential threats arise at both individual and cross-stage levels. The threats related to the RLGMO component are presented in the next chapter.

- **Internal Validity.** These are the internal factors that might affect the results. The comparison to the baseline methods is a likely threat. By using the original source code, testing all associated tools on the same dataset, and comparing them using the same set of metrics, such a danger is diminished.

- **External Validity.** The question of whether our proposed solution is still relevant outside the scope of this work is connected to its generalizability. We expect that the performance of the predictions may be affected by datasets gathered under various circumstances. As a result, training the system with data from various sources will improve its applicability in real-world scenarios.

- **Conclusion Validity.** This relates to our work's experimental conditions, including the simulated setups that were utilized to assess the methodology. We conducted the evaluation using a test set and a training set, which might not accurately reflect real-world usage. We used identical arrangements to compare the systems in order to aim for a fair comparison.

- **Baseline Validity.** We acknowledge that there is some variation in the baseline reference among the models. Consistency may have been affected by variations in pre-processing (such as casing) and the use of a single training run, even if all backbones were evaluated on the same annotated corpus. To guarantee rigorous comparability, all baselines will undergo the same preprocessing in subsequent iterations of this study.

- **Statistical Validity.** We acknowledge that we only used one training run for our trials. Multiple runs under different seeds would provide variance estimates and stronger conclusions, even though this configuration has shown consistent performance increases with fine-tuning across all seven entity classes. Therefore, it is important to carefully evaluate the performance differences between BioClinicalBERT, RoBERTa, and PubMedBERT in Tables 4.8 and 4.9. In order to increase statistical confidence, we want to supplement our findings in future studies using bigger corpora and repeated runs. Although each model was trained for 20 epochs, all experiments were limited to a single run because of computational and dataset constraints. Conducting multiple runs with different seeds or larger-scale tests would allow us to calculate p-values and better estimate variation in the results. This limitation should be considered when interpreting the small performance differences in Tables 4.8 and 4.9, as their statistical significance could change with further experimentation.

- **Pre-processing validity.** To keep only linguistically informative tokens for auxiliary analyses like visualization and word clouds, we kept mostly nouns and adjectives and removed verbs and other functional tokens. The reason was that nouns and adjectives hold the most semantic weight for recognizing entities like diseases, symptoms, and drugs. Yet, this pre-processing operation was not used when training and testing the BERT-based models on full sentence inputs. We did not perform a specific experiment to quantify the impact of this filtering, and future research can compare filtered and unfiltered pipelines to see whether it is beneficial.

Overall, even if these risks draw attention to some of our shortcomings, the reliability of our findings is assured by the consistent experimental design and equitable comparison of models.

4.7.3 Error Propagation to Downstream Modules

Despite the success of the entity extraction models, it is necessary to consider the potential impact of residual errors on the remaining parts of the MetaGRO framework. As the architecture consists of a sequential pipeline of Clinical NER → Knowledge Graph Construction → GCN Embedding → Reinforcement Learning Optimization, any uncertainties introduced in the previous steps could potentially impact the remaining parts of the framework.

First, errors in NER classification will directly impact the construction of the knowledge graph. For instance, if a symptom is mistakenly identified as a disease or if a drug entity is not identified, the constructed graph will have errors in the nodes and edges. Since the knowledge graph represents the structured representation of patient-specific clinical data, any errors in the graph will impact its structure and semantics.

Second, the inaccuracies in the knowledge graph affect the Graph Convolutional Network (GCN) embeddings. The GCNs learn the node embeddings by aggregating information from neighboring nodes based on the graph structure. When the edges are inaccurately represented or there are missing links, the message-passing mechanism will propagate such graph structural errors through layers. Consequently, the embeddings of patients in the latent space might change, possibly affecting the similarity relationships and clustering structures.

Lastly, the Reinforcement Learning-Guided Medicine Optimization (RLGMO) component uses the graph-based embeddings to represent the patient state. When the embeddings are corrupted by previous inaccuracies, the Q-value estimation and action choice might be affected. As the reward function is established based on the similarity between suggested and past successful treatments, the corrupted embeddings could cause the policy updates to be suboptimal during training.

In conclusion, small inaccuracies in entity extraction can propagate through the following chain of errors:

NER error → Knowledge graph distortion → Embedding shift → RL policy variation.

Future work in expanding the MetaGRO framework could take into consideration the usage of uncertainty-aware techniques, such as probabilistic state representation or confidence-weighted graph building, even though the excellent performance of the selected transformer models does mitigate this problem. The MetaGRO framework's transparency and robustness are improved by the explicit acknowledgment of this reliance chain.

4.8 Conclusion

Our research has significantly improved the identification of breast cancer patients by utilizing AI models to transform unstructured clinical data into highly organized, readily retrievable, and comprehensible data. By identifying crucial information like *Age, Gender, Disease, Symptoms, Medications, Doses, and Cancer Stage*, this approach makes it easier for physicians to obtain crucial patient data, which speeds up patient diagnosis and enhances clinical decision-making processes for quicker, more precise, and patient-centered diagnoses and treatments.

In order to efficiently address the complexities involved in the relationships that can be formed between different health conditions, their symptoms, and the associated medication dosages, the next phase of this study focuses on creating medical knowledge graphs and incorporating pattern matching algorithms and Graph Neural Networks (GNNs). This specific phase depends on the work completed in the first phase and creates a logical flow to the entity extraction phase. The work in the applied area of AI-based insights should advance as a result of this extension.

To improve statistical robustness in future experiments, we performed repeated fine-tuning runs across multiple random seeds with detailed reporting of mean performance and variance. This helps address the well-known variability of transformer-based models. Though our study focused on moderately sized runs and limited annotated data due to computational constraints, larger-scale experiments with expanded annotated corpora will be essential to further validate generalization capability.

Furthermore, while our evaluation is based on the labeled clinical dataset curated for this research, comparisons on established biomedical NER benchmarks such as BC5CDR and NCBI Disease remain important future steps. Such benchmarking will help confirm the robustness and external validity of the proposed entity extraction framework beyond the current oncology-focused dataset.

The structured clinical entities extracted in this chapter form the foundation of the patient-specific knowledge graphs described in the next chapter. The reliability of downstream graph embeddings and reinforcement learning–based therapy optimization depends directly on the quality and consistency of this entity extraction process. In this way, the present chapter establishes the essential input layer for the graph-driven reasoning and adaptive therapy recommendation framework introduced in Chapter 5.

Chapter 5

Knowledge Graph–Driven Reinforcement Learning for Therapy Optimization

Artificial Intelligence (AI) has significantly enhanced various clinical applications, including early disease detection, medication recommendation, and surgical planning, in parallel with the rapid advancement of healthcare technologies. Transformer-based models—particularly fine-tuned BioBERT—have demonstrated strong performance in processing medical texts to extract key clinical entities such as diseases, symptoms, and medications. These entities are then organized into a structured tabular format. This work aims to construct knowledge graphs from the structured data to represent relationships between entities. Deeper insights on clinical patterns are then provided by using Graph Neural Networks (GNNs) for analyzing relationships within the graph.

In particular, we used fine-tuned BioBERT from the first part of our work for extracting medical entities from clinical text data [1] related to cancer disease. By fine-tuning BioBERT on our manually annotated corpus, we achieved a high F-score of 97.03%, ensuring reliable extraction of key medical knowledge in our first work. The extracted medical entities are used then to construct knowledge graphs that represent relationships among diseases, symptoms, medications, dosages, and patient histories. These graphs are then trained using Graph Convolutional Networks (GCNs), a subset of Graph Neural Networks (GNNs), to learn meaningful medical patterns. The Reinforcement Learning (RL) model uses the GCN-trained representations as its basic input, allowing it to optimize therapy suggestions according to patient-specific circumstances. Early disease prediction, symptom pattern recognition, and appropriate and individualized therapy planning are all improved by this structured learning pipeline.

However, it should be emphasized that in terms of the entity extraction phase, the knowledge graph and the reinforcement learning module are not distinct parts. In fact, the retrieved items are directly used to represent the patient in the graph embedding of the GCN and RLGMO models. It should be highlighted that the graph topology, the stability of the embeddings, and the recommended behavior of the therapy are all directly impacted by the robustness, uncertainty, and semantic correctness of the NER stage.

In order to recommend suitable treatments, our RLGMO (Reinforcement Learning–Guided Medicine Optimization) model can identify medications that are part

[1] In this work, we used the Summary of Cancer Text dataset that includes unstructured information on cancer types, cancer stages, tumor biomarkers, related diseases, symptoms, medications, treatment responses, medical histories, side effects of chemotherapy, etc.

of the same therapeutic class. The model can recognize the chemical and functional similarities of medications, even when their names vary, and obtains resilient cosine similarity scores of 97.93%. This makes therapy suggestions more adaptable and clinically relevant.

The remainder of this Chapter 5 is structured as follows. Section 5.1 outlines the overall methodology adopted in this study, detailing the framework design, modeling strategies, and integration of graph-based and reinforcement learning components. Section 5.2 presents the data availability (primary and secondary datasets) and describes the clinical datasets used, the entity extraction process, and how the unstructured information was transformed into a structured and interoperable format suitable for knowledge graph construction. Section 5.3 describes the evaluation process, including dataset partitioning, reward computation logic, and validation strategies used to assess model performance and reliability. Then next, Section 5.4 presents and interprets the experimental results, highlighting the model's ability to generate personalized, clinically consistent medication recommendations. Section 5.5 provides a comprehensive discussion of the findings, implications, and limitations, relating them to the research objectives and broader clinical context. Finally, Section 5.6 concludes the thesis by summarizing the contributions, emphasizing the impact of the proposed framework, and outlining future directions for extending the system toward real-world clinical decision support applications.

5.1 Methodology

This section presents the research methodology adopted to develop the proposed RLGMO module. Section 5.1.1 describes the hardware configuration and computational setup used in this study. Section 5.1.2 explains the construction of a structured knowledge graph connecting patients with symptoms, diseases, medications, and cancer stages. Section 5.1.3 outlines the application of Graph Convolutional Networks (GCNs) to extract informative patient embeddings from the graph topology. Section 5.1.4 details the reinforcement learning framework, including the agent design, definition of state and action spaces, and the dual-source reward mechanism that integrates historical similarity and clinical guideline validation. Finally, Section 5.1.5 describes the evaluation protocol, including test data formulation and performance metrics used to assess the therapy recommendation model.

5.1.1 Data Preprocessing

To prepare the datasets for training and evaluation, two data sources were preprocessed: (i) the primary clinical dataset used to construct patient knowledge graphs and (ii) a reference dataset used exclusively for reward validation in the reinforcement learning framework.

The unstructured clinical text data collected in the earlier phase was structured through entity extraction, where medical entities such as diseases, symptoms, medications, doses, and medical history were identified and organized into a structured format. Among the evaluated models, **BioBERT** demonstrated the best performance in extracting these entities, and therefore, its extracted results were used in the subsequent GNN and RLGMO stages. The extracted and structured entities were stored in a `CSV` format to promote better understanding, reproducibility, and interoperability for downstream processing.

The main clinical dataset contained structured information such as age, symptoms, diseases, medications, doses, medical history, and cancer stages. We cleaned this dataset by standardizing column names, removing incomplete entries and duplicates, and normalizing text fields (e.g., lowercasing, punctuation removal). We also harmonized entity terms across symptoms, diseases, and drugs using vocabulary matching. From the cleaned data, patient-level knowledge graphs were constructed by linking each patient node to their associated medical entities.

In contrast, the reference dataset was not used in the graph construction. Instead, it provided structured diagnosis–treatment mappings to validate recommendations made by the RL agent. Entity names from this dataset were aligned with the clinical vocabulary using fuzzy string matching. Both datasets were aligned under a common schema to ensure consistency during training and evaluation, particularly in supporting dual-source reward computation.

Hardware Settings

All experiments in the Reinforcement Learning Guided Medicine Optimization (RLGMO) were conducted on a high-performance standalone workstation with the following specifications:

System	High-performance standalone workstation
Operating System	Windows 11 Pro
CPU	AMD Ryzen 9 5950X @ 3.40GHz, 16 cores / 32 threads
GPU	NVIDIA RTX 3090 with 24 GB GDDR6X VRAM
RAM	128 GB DDR4
Storage	2 TB NVMe SSD
Frameworks and Libraries	PyTorch 2.1 HuggingFace Transformers 4.40+ PEFT (LoRA support) PyTorch Geometric for GNN modeling Stable-Baselines3 for reinforcement learning
Environment Management	Anaconda (Python 3.10) with isolated virtual environments

High memory bandwidth and parallel processing capabilities made this configuration ideal for large-scale biomedical workloads, allowing for the effective training and evaluation of deep learning models, such as GNN-based patient representations and reinforcement learning agents.

5.1.2 Knowledge Graph Construction

The second step in the RLGMO pipeline consists of building an RLGMO-level knowledge graph, as shown in Figure 5.1. The graph represents structured relationships among various clinical entities extracted during preprocessing—symptoms, diseases, medications, doses, and cancer stages. Each node representing a patient is connected by edges to the corresponding clinical terms. Node labels with type specifications indicate the presence of semantic meanings for these entity types (e.g., `symptom`, `disease`, `medication`, etc.).

Next to RLGMO-level subgraphs, an auxiliary set of nodes from the reference dataset is introduced, which codifies the external diagnosis–treatment mappings.

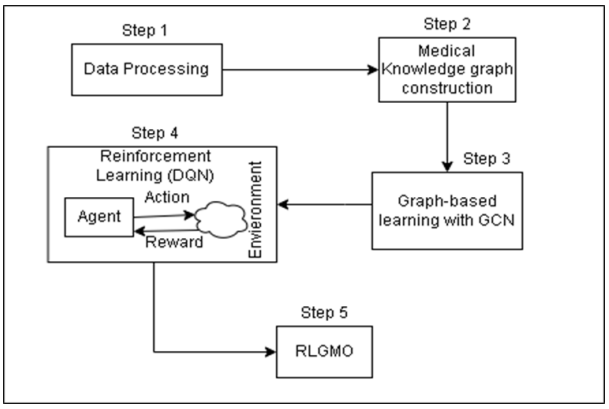


FIGURE 5.1: Our proposed RLGMO pipeline step by step

These are connected to `reference` nodes that link diagnosis terms to treatments recommended in the reference. The incorporation of such a structure enriches the graph with a consideration of connections within patients and across guidelines, setting the final graph structure ready for learning and reward validation. The final output of this phase is a heterogeneous knowledge graph that encapsulates complex clinical relationships, forming the foundation for graph learning and RL-based treatment optimization.

5.1.3 Graph-Based Learning with GNNs

The next step in the RLGMO pipeline is to use Graph Neural Networks (GNNs) to learn expressive patient embeddings after the knowledge graph construction. The graph structure, which is constructed of different clinical nodes and their connections, is converted into a format appropriate for GNN-based representation learning, as shown in Figure 5.1.

We capture local neighborhood structures for every patient node by using a Graph Convolutional Network (GCN) to propagate and aggregate information across the graph. These embeddings use common symptoms, diseases, medications, and stages of treatment to encode semantic similarity between patients. A dense vector representation for every patient node is the step's output, and it will be used as the input state by the reinforcement learning agent in the next step. The model can reason about contextual relationships and patient similarity in a data-driven and scalable way because of these learned embeddings.

5.1.4 Reinforcement Learning for Therapy Optimization

We use Reinforcement Learning (RL) in the last optimization step of the RLGMO in Figure 5.1 to suggest treatments based on patient representations that have been learned. The RL agent engages with a personalized environment in which actions stand in for possible medications and each state relates to a patient embedding produced by the GCN. When an action is selected, the agent is rewarded using a dual-source logic that includes *(i)* alignment with past treatment outcomes for similar patients and *(ii)* validation against a reference dataset of recommendations based on guidelines.

We employ a Deep Q-Network (DQN) that learns a policy to maximize long-term reward as part of a model-free Q-learning approach. With this configuration, adaptive learning is possible for a variety of RLGMO profiles. Clinical restrictions,

such as partial matches and RxNorm-based drug equivalency, are represented in the environment. The agent learns to make recommendations for treatments that are both historically and clinically sound as it develops its decision-making process.

5.1.5 Evaluation Design for RLGMO

A proprietary reward-based evaluation technique, specifically adapted for the clinical domain, was used to assess the RLGMO module. Rather than relying solely on traditional evaluation metrics, we focused on the alignment between recommended medications and real-world prescriptions. The evaluation employed a reward logic that captures similarity patterns across patient history, symptoms, and diseases. In addition, cosine similarity was applied to measure the closeness between suggested and actual medications, ensuring that recommendations reflect both patient-specific patterns and clinical relevance.

Furthermore, we used drug ontology mappings, that is RxNorm API[2], to ensure clinical validity by conducting class-equivalent checks to verify that the suggested treatments belonged to the same therapeutic class of drugs that were prescribed. This thorough assessment made sure that our system gave both medical relevance and personalization top priority when making recommendations.

5.2 Data Availability and Preprocessing

For the second stage of this study, which focuses on building the RLGMO model, we draw upon the same clinical dataset introduced earlier in the entity-extraction chapter, Section 4.1.1. The original corpus consists of 40 breast cancer patient records curated by Sushil et al. [132] from the UCSF Information Commons, covering the period 2012–2022. All records are fully de-identified, include detailed oncology notes, and provide complete staging information, disease progression, and treatment history. The dataset is rich in clinical detail—containing thousands of annotated entities, modifiers, and relationships—which makes it suitable for downstream graph-based modelling.

Dataset 1: (Primary dataset)

In the context of this work, however, the primary input to the RLGMO component is not the raw UCSF corpus but the **structured entities extracted using the fine-tuned BioBERT model**. These entity outputs—covering *diseases, symptoms, medications, doses, cancer stage, age, and gender*—form the core dataset used for constructing patient-specific knowledge graphs and for learning optimal treatment policies in the reinforcement-learning stage.

Dataset 2: (Reference dataset)

To support therapy validation in our RLGMO model, we incorporated a reference dataset introduced in the work of P. Larmuseau, a pharmacist at Gaver Apotheek, Belgium [133]. The original dataset[7]—used in their published system—comprises eight structured CSV files containing disease-symptom mappings, treatment courses, and weighted associations. While this dataset was not suitable for direct training

[2]RxNorm provides normalized names for clinical drugs and links its names to many of the drug vocabularies commonly used in pharmacy management and drug interaction software, including those of First Databank, Micromedex, Multum, and Gold Standard Drug Database

due to its general scope and synthetic nature, it served as an important external knowledge base to cross-check and validate treatment recommendations derived from our primary clinical dataset. Some of the CSV files are listed below:

1. `sym_t.csv`: $(syd, symptom)$ = (Symptom identifier, Symptom name)
2. `dia_t.csv`: $(did, diagnose)$ = (Disease identifier, Disease name)
3. `diffsydiw.csv`: (syd, did, wei) = (Symptom identifier, Disease identifier, Weight of the symptom on the disease)
4. `prec_t.csv`: $(did, diagnose, pid)$ = (Disease identifier, Disease name, Treatment course)

In our model, this dataset was referred to as the *reference dataset*. Specifically, we used the disease-treatment mapping (`prec_t.csv`)[8] and the symptom-disease weight file (`diffsydiw.csv`) to support reward computation in the reinforcement learning environment. This reference dataset of these 2 CSV files consists of a total of 1,167 diseases and 273 symptoms. This allowed our agent to align therapy suggestions not only with learned historical patterns but also with curated treatment guidelines derived from real-world pharmaceutical expertise. All reference files were cleaned by removing NULL values and harmonizing delimiters, following the original authors' preprocessing steps. This hybrid setup ensured a more clinically grounded and generalizable recommendation system.

5.3 Evaluation Process details

Our approach constructs knowledge graphs to discover hidden patterns between diseases, symptoms, and medications in patients' history records. These structured graphs are then used with Graph Neural Networks (GNNs) to learn meaningful patient representations. Finally, we train a reinforcement learning (RL) agent to optimize personalized therapy recommendations based on learned treatment strategies.

The primary goals of this research are:

- To develop a pipeline for the extraction of medical entities using a fine-tuned BioBERT model and training of knowledge graphs driven by GNN. By identifying hidden clinical links between diseases, symptoms, and medications, this combination improves the ability to recognize patterns in cancer treatment.
- To implement a hybrid AI approach that unifies graph-based representation learning (GNN) and reinforcement learning (RL) for treatment decision-making —thus enabling automated, data-driven, and patient-specific therapy recommendations.
- To train a reinforcement learning (RL) agent on structured knowledge graph data to enable iterative improvement in therapy suggestions by allowing it to investigate treatment alternatives that minimize mismatches and optimize clinical relevance.

[6] `https://physionet.org/content/curated-oncology-reports/1.0/`
[7] `https://www.kaggle.com/plarmuseau/sdsort`
[8] `https://github.com/sud0lancer/Diagonosis-Precaution-dataset`

By achieving these objectives, our research work aims to improve the efficiency of medical data processing and promote AI-assisted clinical decision-making, particularly for cancer patients. No previous research has thoroughly investigated several techniques based on patient similarity and knowledge graph representations to determine the best approach for optimizing drug recommendations using reinforcement learning. In order to close this gap, we present and assess the Reinforcement Learning Guided Medicine Optimization (RLGMO) in the MetaGRO framework. We provide a thorough review of how RLGMO uses organized medical data to improve clinical decision-making. In this chapter, here are the following study in this respect:

1. **Knowledge Graph Construction for Pattern Discovery:** We construct the knowledge graphs from the extracted medical entities that connect patients to clinical concepts. By detecting hidden patterns and correlations, graph-based learning using GNNs can improve our comprehension of the medical context.
2. **Development of RLGMO for Therapy Optimization:** We propose RL-GMO (Reinforcement Learning Guided Medicine Optimization), a novel hybrid model that integrates GCN embeddings and a Deep Q-Network (DQN) for learning optimal treatment strategies. This approach supports personalized care by adapting over time to patient data.

While prior studies have explored knowledge graph construction and AI-assisted diagnosis [134], they often remain limited to static visualizations or summary tools. Our work addresses this limitation by introducing a dynamic, pattern-matching framework that integrates structured knowledge graphs, graph neural networks (GNNs), and reinforcement learning (RL) to support adaptive clinical decision-making.

In cancer care—particularly breast cancer—therapeutic planning is complicated by the evolving nature of symptoms during treatment. Our proposed RLGMO model captures these dynamic changes by correlating historical patient data with new clinical observations, enabling more personalized and evidence-based therapy recommendations.

The following primary research challenge serves as the framework for our contributions:

How can we design an interpretable and adaptive AI module that integrates structured patient data, knowledge-graph-based learning, graph neural networks, and reinforcement learning to generate clinically relevant medication recommendations?

To address this challenge, we propose a novel therapy recommendation framework that combines structured pharmaceutical mappings with clinical data. The constructed knowledge graph connects RLGMO_IDs with diseases, symptoms, medications, and cancer stages. RLGMO embeddings are generated using Graph Convolutional Networks (GCNs) and used to guide a Deep Q-Network (DQN)-based model-free reinforcement learning agent. A key innovation of our approach is the dual-reward mechanism, which evaluates medication suggestions based on both patient similarity and formal treatment guidelines—ensuring they are both personalized and clinically relevant. Additionally, the framework offers real-time recommendations with graph-based interpretability, setting it apart from conventional black-box AI models.

Our research is guided by the following questions:

- **RQ1: How can structuring clinical records into a knowledge graph reveal interpretable patterns in oncology care and improve data interoperability?**

  This research question explores the development and role of a structured knowledge graph in transforming already structured clinical data into a meaningful and machine-interpretable representation. The dataset—derived from processed clinical records stored in CSV format—includes entities such as symptoms, disease progression, diagnostic outcomes, treatment histories, and medication patterns. By systematically organizing this information into a knowledge graph, the framework establishes a foundation for uncovering hidden relationships and recurring patterns within oncology data.

  The investigation focuses on how graph-based modeling can identify correlations between symptoms and specific cancer types, track progression trends across diverse patient profiles, and evaluate treatment effectiveness. Through graph-based pattern recognition, the study aims to enhance clinical understanding, improve predictive insights, and support evidence-driven decision-making in oncology care. Ultimately, this question demonstrates that structuring clinical records into a knowledge graph not only improves data organization and interoperability but also provides a robust analytical framework to advance personalized medicine and patient (RLGMO) outcome prediction in cancer research.

- **RQ2: How can graph-based representations and similarity measures derived from patient knowledge graphs be used to identify effective therapy strategies from historical oncology outcomes?**

  This research question explores the identification and recommendation of effective therapy strategies using graph-based representations of clinical data when combined with similarity computation algorithms. Relationships between past cases can be thoroughly analyzed by describing patients' illnesses, symptoms, and therapies as linked nodes within a knowledge graph. The use of similarity measures—such as cosine similarity or graph-based distance metrics—enables the identification of patients with comparable clinical profiles, treatment histories, and outcomes.

  The objective is to discover how these graph-based models can learn from past cancer cases in order to produce treatment recommendations that take into account actual clinical trends and data. This method facilitates a data-driven decision-making process by comparing the profiles of new patients with those of previous instances that show comparable disease trajectories and treatment responses. The ultimate goal of this research question is to find out how combining graph-based similarity learning with past outcome data can support more accurate, individualized, and flexible treatment recommendations in oncology care.

- **RQ3: How can reinforcement learning improve therapy recommendations using knowledge graph–based patient representations for new oncology cases?**

  This research question addresses how reinforcement learning (RL) can be applied to optimize therapy selection in oncology by using the insights derived from knowledge graph–based representations of patient (RLGMO) data. By integrating patient (RLGMO)-specific features, disease characteristics, and therapy outcomes within a graph-structured framework, the RL model can learn

adaptive strategies that continuously improve through interaction and feedback.

This part of the study mainly focuses on how the RL agent can identify optimal treatment pathways by analyzing patterns and similarities among patients embedded in the knowledge graph. Through the iterative learning, the agent refines its decision-making process, balancing exploration of new therapeutic possibilities with the exploitation of successful historical strategies. The model can adapt dynamically because of its adaptive learning technique, providing context-aware and customized treatment recommendations.

Ultimately, this question seeks to determine how reinforcement learning can bridge structured medical knowledge with real-world clinical adaptability, ensuring that therapy recommendations are both data-driven and responsive to individual patient needs in oncology care.

5.4 Evaluation Results

For the RLGMO stage, we do not reuse the raw CORAL clinical notes directly; instead, we rely on the structured entity outputs obtained from the BioBERT-based extraction pipeline described in the previous chapter. These extracted entities—covering diseases, symptoms, medications, doses, cancer stage, age, and gender—serve as the primary dataset for building patient-specific knowledge graphs and learning treatment policies.

To supplement the limited size of the CORAL dataset and to incorporate broader medical domain knowledge, we additionally employed an external tabular dataset[3] introduced by Singh Punn et al. [133]. This dataset contains structured mappings between symptoms, diseases, and recommended remedies across multiple medical conditions. Although not cancer-specific, it provides a useful reference for modeling general disease–symptom–treatment associations.

Together, the extracted clinical entities and the supplementary tabular knowledge form the basis for constructing the healthcare knowledge graph required by the RLGMO module. This graph encodes relationships such as *patient* $\rightarrow$ *disease* $\rightarrow$ *medication* and *patient* $\rightarrow$ *symptoms* $\rightarrow$ *medication*, enabling downstream graph-based reasoning for personalized treatment recommendations.

This section reports and analyzes the obtained results by answering the key research questions:

5.4.1 RQ1: How can structuring clinical records into a knowledge graph reveal interpretable patterns in oncology care and improve data interoperability?

As introduced in the preliminaries, we represent each patient's clinical profile as a structured graph $G_i = (V_i, E_i)$, where nodes correspond to key medical entities and edges capture their relationships. To build these graphs [135] from raw data, we used the `NetworkX` library in Python, which allows for flexible construction, manipulation, and visualization of complex networks.

Each patient is represented by a unique node that connects to various associated entities, including symptoms, diagnosed diseases, prescribed medications, dosage

[3]Disease Diagnosis and Recommended Remedy dataset, available on IEEE DataPort: `https://ieee-dataport.org/documents/disease-diagnosis-and-recommended-remedy`.

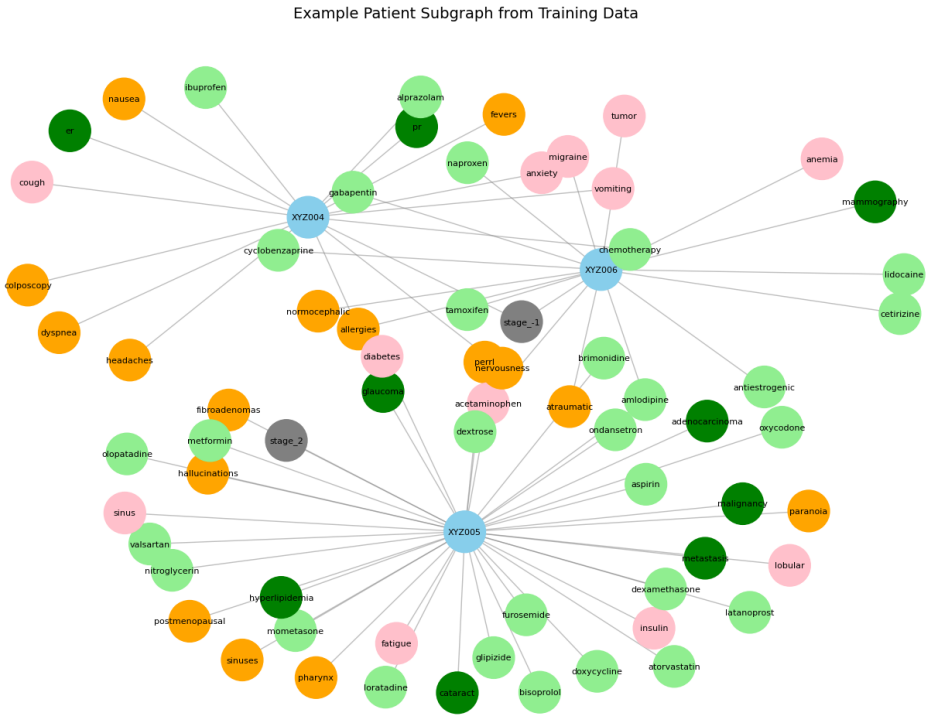


FIGURE 5.2: knowledge subgraph between 3 connected patients with RLGMO_ID: skyblue, Symptoms: orange, Disease: green, Medication: light green, stage: gray, Diagnose:pink

information, and cancer stages. These connections are captured through undirected edges, creating a semantic structure around the patient's condition. For example, a patient (RLGMO) node may connect to one or more disease nodes via *has_disease* edges, and to medications via *treated_with* edges. This design enables the aggregation of medical patterns across different cases and supports downstream tasks such as recommendation and similarity matching.

To streamline the construction process, we generalized the logic by dynamically looping through core entity types such as `Symptoms`, `Disease`, `Medication`, and `Dose`, and conditionally linking them to each patient node. When available, cancer stage data is added as a node (e.g., `stage_2`) and connected as well.

In addition to individual patient data, we incorporated expert-curated reference guidelines. Each reference rule is represented as a node (e.g., `REF_id`) and connected to corresponding diagnosis terms and their suggested treatment. These nodes form an external knowledge source that complements the primary data and allows the model to match patient states against generalized clinical protocols. This Figure 5.3 represents how RLGMO uses the disease-based connections between patients to identify treatment patterns, like the other symptom-based connections. Each link shows shared clinical terms, enabling the reinforcement learning model to match patients (RLGMO) and guide therapy decisions effectively.

This structured representation is summarized in Algorithm 8. It serves as the backbone for visual analytics and graph-based learning models in our pipeline.

Algorithm 8 describes how we construct the healthcare knowledge graph G from the structured cancer clinical dataset $\mathcal{D}$ and the external reference guidelines $\mathcal{G}$. The graph is initialized as empty and then populated with nodes and edges corresponding to patients and their clinically relevant attributes. For each RLGMO record

FIGURE 5.3: The overview of pattern matching in diseases using a knowledge graph.

$r \in \mathcal{D}$, we first create a node for the patient (RLGMO), identified by `RLGMO_ID`, and conceptually labelled as a `Patient (RLGMO)` node. We then iterate over the main clinical entity types—`Symptom`, `Disease`, `Medication`, and `Dose`—and, for each non-empty value v listed in the record, we add a node of the corresponding type and connect it to the RLGMO node with an edge. In a similar way, if a cancer stage is available, a `Stage` node is created and linked to the patient (RLGMO), and any entries in the `Medical_History` field are treated as additional symptom-like nodes, also connected back to the same patient. In this way, each row in the structured RLGMO dataset becomes a small, labelled subgraph that encodes the relationships between a patient and their symptoms, diagnoses, medications, doses, stage, and history.

The second part of the algorithm integrates external clinical guidelines into the same graph. For each guideline triple $(\mathcal{T}, t, id) \in \mathcal{G}$, we create a `Reference` node with identifier `REF_`id. This node is then connected to all terms $x \in \mathcal{T}$ that appear in the guideline (for example, specific diseases, symptoms, or medications) and optionally linked to a target concept t when it is defined (such as a recommended treatment or protocol). These `Reference` nodes act as labeled anchors for medical knowledge and link together related clinical concepts across patients. The resulting knowledge graph $G = (V, E)$ therefore combines RLGMO-level information from the RLGMO records with guideline-level relations, providing a structured, type-labeled representation that can be used by downstream GNN and reinforcement learning components for context-aware treatment reasoning.

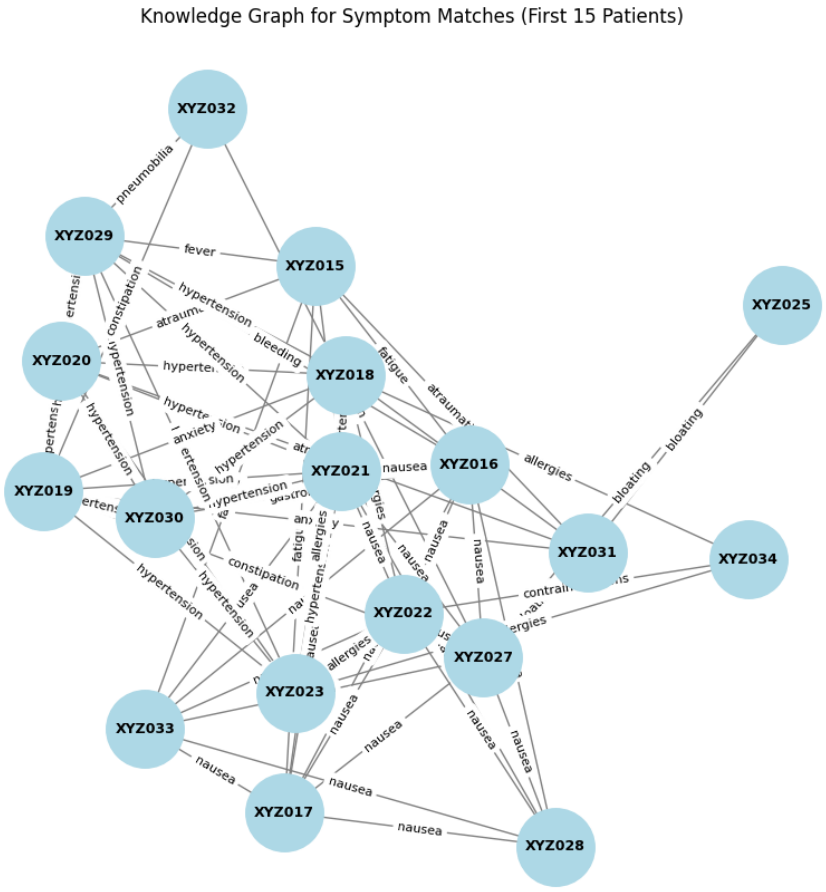


FIGURE 5.4: The overview of pattern matching in symptoms using knowledge graphs.

Answer to RQ$_1$. Key clinical entities—including diseases, symptoms, medications, and cancer stages—were extracted using a fine-tuned BioBERT model from unstructured clinical records in the previous stage. These extracted entities were then organized into a structured knowledge graph, forming the foundation of the RLGMO framework. The knowledge graph enables a relational and interpretable representation of oncology RLGMO profiles, facilitating the discovery of recurring treatment patterns and supporting data interoperability for downstream analysis.

5.4.2 RQ2: How can graph-based representations and similarity measures derived from patient knowledge graphs be used to identify effective therapy strategies from historical oncology outcomes?

After constructing the knowledge graph, we train a Graph Convolutional Network (GCN) to learn meaningful embeddings for RLGMO nodes based on their clinical context. Each patient node is associated with a feature vector consisting of demographic and clinical attributes: age, gender, and cancer stage. These features are normalized to ensure uniform contribution during training.

We implement a two-layer GCN using PyTorch Geometric. The first GCN layer projects the input features to a hidden space of 16 dimensions, followed by ReLU activation. The second layer maps them into a 3-dimensional latent embedding space.

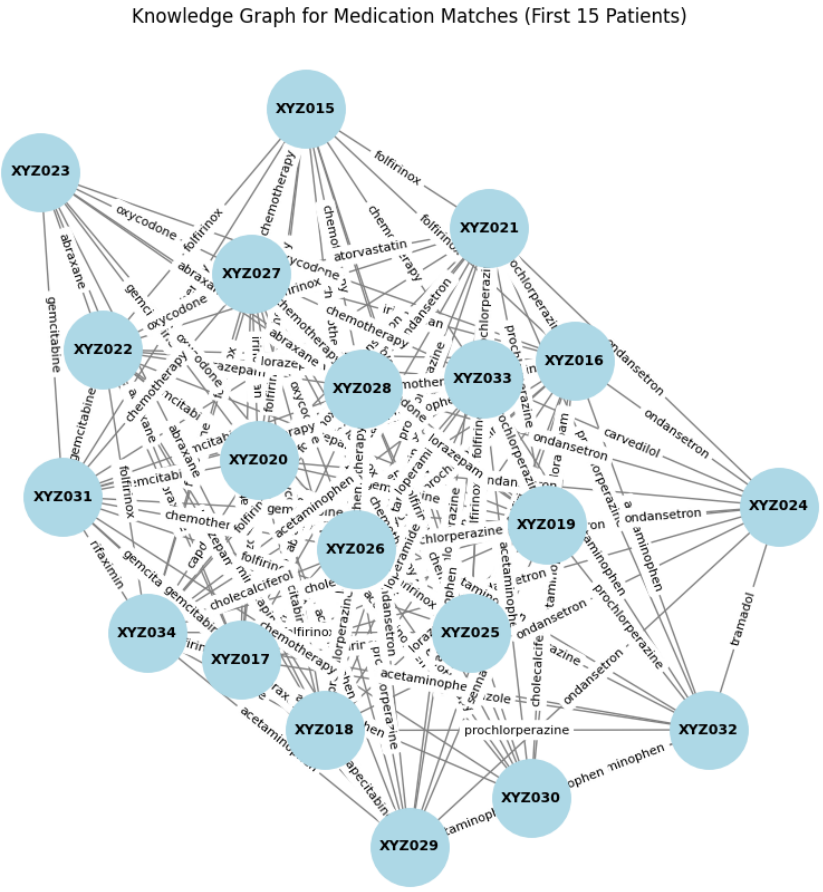


FIGURE 5.5: The overview of pattern matching in medications using knowledge graphs.

We use the Mean Squared Error (MSE) loss between the reconstructed output and the original input features for unsupervised training. The model is trained for 2000 epochs using the Adam optimizer with a learning rate of 0.005.

A simplified excerpt of the training code is as follows:

```
class GCNEncoder(nn.Module):
    def __init__(self, in_channels, hidden_channels, out_channels):
        super().__init__()
        self.conv1 = GCNConv(in_channels, hidden_channels)
        self.conv2 = GCNConv(hidden_channels, out_channels)

    def forward(self, x, edge_index):
        x = self.conv1(x, edge_index).relu()
        return self.conv2(x, edge_index)
```

The embeddings generated by this model are extracted for RLGMO nodes and stored for downstream reinforcement learning tasks. A visualization of the training loss across epochs confirms convergence and stability.

Algorithm 9 outlines the training procedure of the GCN used to generate node embeddings for the RLGMO module. The input feature matrix X contains the structured attributes extracted for each RLGMO instance—such as symptoms, diseases, medications, doses, and numerical fields like age or cancer stage—while the edge list E encodes the relationships defined in the underlying medical knowledge graph.

Prior to training, the selected numerical features X are normalized, and the edge list is converted into an adjacency or edge-index representation A compatible with graph operations. The GCN parameters θ are then initialized based on the input, hidden, and output dimensions.

Training proceeds for T epochs. In each epoch, a forward pass computes node embeddings $Z = \text{GCN}(X, A; \theta)$ by aggregating information from neighboring RLGMO nodes within the graph structure. A self-supervised reconstruction loss, implemented here using a mean squared error objective $\text{MSE}(Z, X)$, encourages the learned embeddings to preserve the key characteristics of each RLGMO instance. Gradients are obtained through backpropagation, and the optimizer updates the parameters θ accordingly. After training, the final embeddings Z_p serve as the representation of each RLGMO node, which are then used as input for the subsequent reinforcement learning component. Both the embeddings and the index mapping for the RLGMO instances are stored for use in later stages of the pipeline.

To evaluate how well the model learned over time, the Mean Squared Error (MSE) loss was recorded across 2000 training epochs. As shown in Figure 5.6, the curve starts with a higher loss and gradually decreases before flattening out, which indicates that the model improved steadily until it reached a stable point. This consistent decline confirms that the model successfully learned useful data patterns, resulting in reliable embeddings for further analysis.

To better understand how the model represents RLGMOs, we used two visualization methods—t-SNE and UMAP—to project the high-dimensional GCN embeddings into a two-dimensional space. As shown in Figure 5.8, these visualizations help reveal how RLGMOs are grouped according to similarities learned by the model.

The t-distributed Stochastic Neighbor Embedding (t-SNE) technique reduces complex, high-dimensional data into two or three dimensions by emphasizing the preservation of local relationships. In other words, it places similar data points close together and separates dissimilar ones, making local patterns visible even when global structure is less clear.

The Uniform Manifold Approximation and Projection (UMAP) method also reduces dimensionality but focuses on maintaining both local and global relationships within the data. It builds a graph representation of the dataset and then optimizes its low-dimensional projection to retain meaningful neighborhood structure.

Subfigure 5.8a shows the t-SNE projection, where patient (RLGMO) points appear more scattered, highlighting local variations among individuals. Subfigure 5.8b displays the UMAP projection, which reveals more clearly defined clusters of related RLGMOs. Finally, Subfigure 5.8c colors each point according to cancer stage, allowing visual comparison of RLGMOs with similar clinical progressions.

Overall, these visual patterns suggest that the GCN effectively captures clinically meaningful relationships among RLGMOs, demonstrating its potential for downstream applications such as identifying similar cases or recommending appropriate therapies.

By analyzing the learned graph structure, we identified several triplets derived from the knowledge graph, such as (`RLGMO-diagnosis-medication`). These examples were obtained by examining how connected nodes cluster after GCN embedding, revealing recurring and meaningful relationships between diseases and prescribed treatments across different RLGMOs.

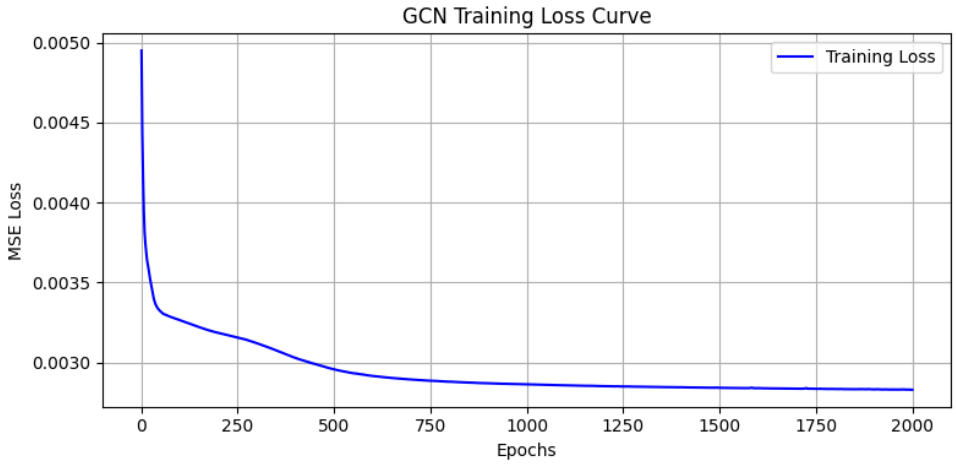


FIGURE 5.6: GCN Training MSE loss VS. Epochs

Top 16 GCN-Derived Clinical Patterns

XYZ001 -> hernia -> tylenol
XYZ002 -> stroke -> rosuvastatin
XYZ003 -> prostate -> iron
XYZ004 -> asthma -> hypokalemia
XYZ005 -> stroke -> doxycycline
XYZ006 -> asthma -> hypokalemia
XYZ007 -> thyroid -> dexamethasone
XYZ008 -> asthma -> hypokalemia
XYZ009 -> endometriosis -> amoxicillin
XYZ010 -> pancreas -> iron
XYZ011 -> asthma -> hypokalemia
XYZ012 -> hernia -> clonazepam
XYZ013 -> hernia -> clonazepam
XYZ014 -> hernia -> tylenol
XYZ015 -> thrombocytopenia -> enoxaparin
XYZ016 -> thrombocytopenia -> tylenol

As shown in the above box, the presented GCN-derived clinical patterns are generated from the context of *Breast Cancer* and *Pancreatic Cancer* patients (RLGMOs). These associations, while meaningful within the cancer treatment domain, may not reflect typical patterns observed in healthy individuals or in patients without malignancies. Therefore, this study is confined to cancer-related diagnoses and treatments, with the goal of identifying treatment-relevant correlations between illnesses, symptoms, and prescription drugs.

However, the same graph-based approach can be extended to other medical conditions beyond cancer, enabling pattern discovery for non-oncological diseases and symptoms in the general population. This flexibility demonstrates the adaptability of the proposed GCN framework to a broader range of healthcare datasets and diagnostic contexts.

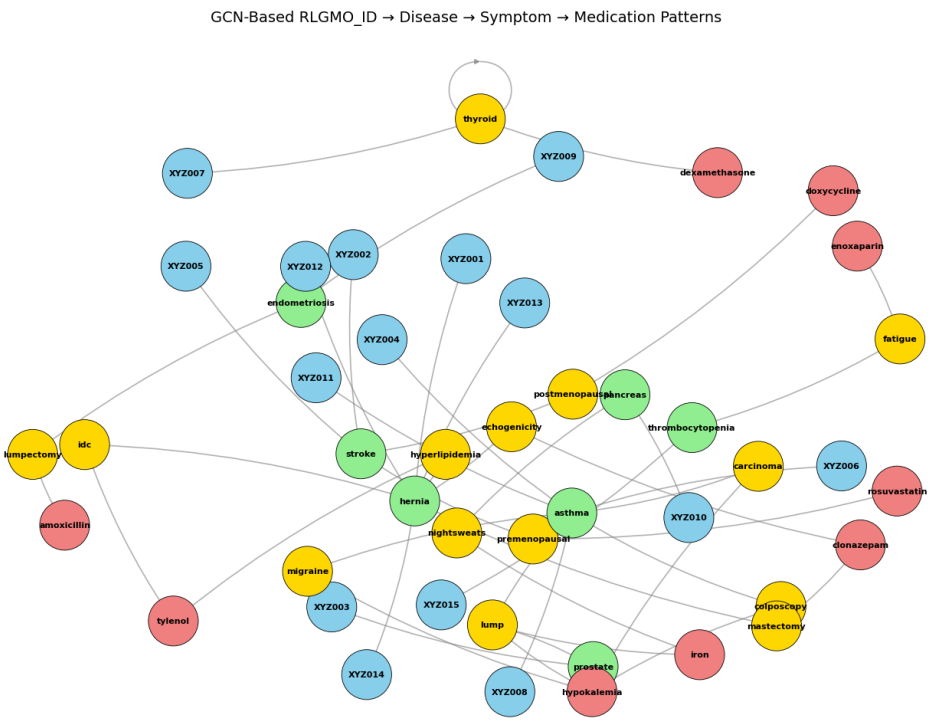
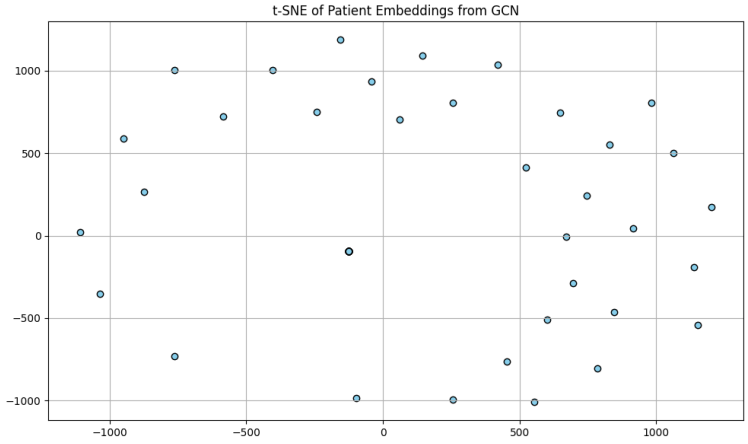
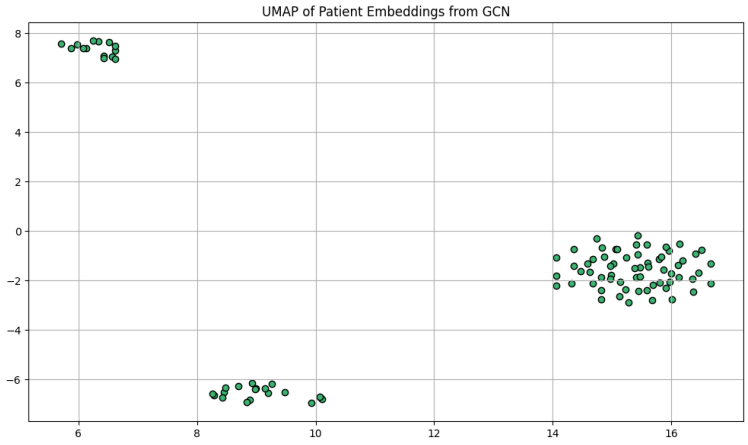


FIGURE 5.7: Patterns among RLGMO_IDs with disease, symptoms, and medication patterns

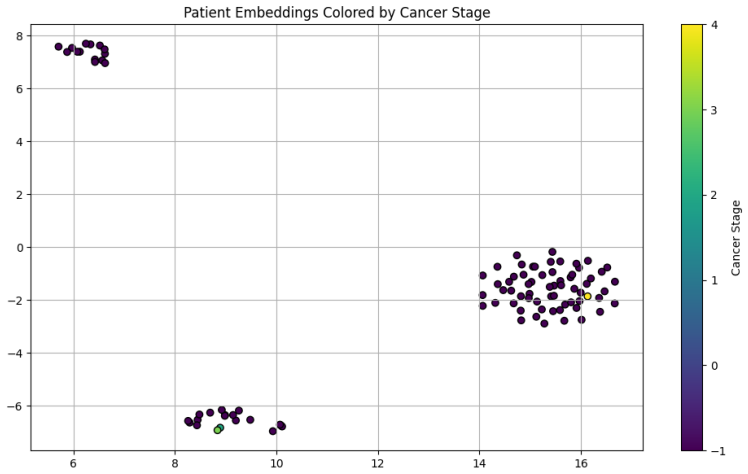
Answer to RQ$_2$. Graph Convolutional Networks (GCNs) were employed to learn patient embeddings that encode both structural and semantic relationships within the knowledge graph. These embeddings enable the identification of patients with comparable clinical profiles, capturing correlations among diseases, symptoms, and medications. By analyzing these graph-based similarities, the system effectively identifies therapy strategies aligned with successful historical outcomes, supporting evidence-driven and interpretable treatment recommendations.



(A) t-SNE of RLGMO embeddings.



(B) UMAP of RLGMO embeddings.



(C) UMAP colored by cancer stage.

FIGURE 5.8: Visualization of GCN-generated RLGMO embeddings using dimensionality reduction techniques. Subfigure (a) shows a t-SNE projection, (b) shows UMAP without labels, and (c) colors the UMAP points based on cancer stage labels. These plots demonstrate the clustering structure learned by the GCN model.

5.4.3 RQ3: How can reinforcement learning improve therapy recommendations using knowledge graph–based patient representations for new oncology cases?

To simulate the recommendation process and reward evaluation, we implemented a custom OpenAI Gym-compatible environment. The agent receives an RLGMO embedding as the observation and selects a medication from the available action space. The reward is based on RLGMO similarity, correct medication match, drug class equivalence (via RxNorm), or reference guideline support.

Algorithm 11 describes the core reinforcement learning step implemented in the `MedicalRecommendationEnv` environment, where the agent proposes a medication for a given RLGMO instance. The environment takes as input the selected action a and the current RLGMO identifier p. First, it retrieves the GNN-based embedding vector v_p for RLGMO p and maps the action a to its corresponding medication m. It then gathers the disease, symptom, and medication sets associated with p.

The reward is computed in two stages. In the first stage, the environment searches for another RLGMO $q \in \mathcal{P}$, $q \neq p$, that (i) has a valid embedding, (ii) shares at least one disease or symptom with p, and (iii) includes the candidate medication m in its medication list. If such a similar RLGMO is found, the cosine similarity between v_p and v_q is calculated and combined with the number of shared clinical terms to produce a reward:

$$r = \min\big(1.0,\ 0.6 + 0.1 \times |\mathit{shared\_terms}| + 0.3 \times \mathit{sim}\big),$$

and the match type is labelled as `similar_RLGMO`. This encourages the agent to recommend medications that have worked for clinically similar RLGMO cases in the past.

If no similar RLGMO is found, the environment falls back to a rule-based hierarchy grounded in the original data and external knowledge sources. If the proposed medication m appears among the actual medications of p, a reward of 0.8 is given and the match is tagged as `exact`. If m does not appear directly but is class-equivalent via RxNorm or is supported by a reference guideline, a moderate reward of 0.6 is assigned with match type `class_equivalent` or `guideline`, respectively. If neither of these applies but some disease or symptom context is present, a lower reward of 0.3 is given and the match is marked as `partial`. In cases where there is no useful information to support or contextualize the recommendation, the reward is set to 0.0 with the match type `no_info`. The step then returns the next state, the computed reward r, a terminal flag `done = true` (reflecting a one-step interaction), and an `info` object that records the match type and other diagnostic details.

The reinforcement learning agent is trained using the Deep Q-Network (DQN) algorithm from the Stable-Baselines3 library. The training leverages RLGMO-specific embeddings generated by the GCN and integrates them into a custom Gym environment named `MedicalRecommendationEnv`. This environment uses both historical patient data and expert reference guidelines to evaluate the quality of medication recommendations.

The training dataset, loaded from `extracted_annotations_cleaned.csv`, contains medical entities such as symptoms, diseases, and prescribed medications for each patient. These entities are parsed and structured into dictionaries keyed by RLGMO ID. Medications are lowercased for uniformity.

The medical recommendation environment is constructed using:

- GNN embeddings for each patient from `RLGMO_gnn_embeddings.pt`.

- A node map linking RLGMO IDs to embedding indices.
- A precompiled list of medications from `med_list.pkl`.
- Reference diagnosis-to-medication mappings from `Patient_data.csv`.

To facilitate reliable learning and checkpointing, the environment is wrapped with a `Monitor` and a `DummyVecEnv`, and a `CheckpointCallback` is used to periodically save model checkpoints.

The DQN model is initialized with a multilayer perceptron policy and several optimized hyperparameters:

Hyperparameters for RL Model Training

- **Learning rate:** 1×10^{-4}
- **Buffer size:** 50,000 transitions
- **Batch size:** 64
- **Discount factor (γ):** 0.99
- **Target update interval:** 500 steps
- **Exploration fraction:** 0.2 with final $\epsilon = 0.05$
- **Train frequency:** 1 step
- **Device:** CUDA if available, else CPU

The training process runs for 5,000 timesteps, with logs written to TensorBoard and models saved to `trained_rl_model`. This setup ensures traceability and reproducibility of the training experiment.

Algorithm 10 outlines the Deep Q-Learning procedure used to train the therapy recommendation agent in our RLGMO-based environment. The training begins by initializing the replay buffer and the Q-network, while the target network is set as a copy of the initial parameters to ensure stable learning. For every episode, the environment is reset and an initial state s_0 is generated. At each step, the agent follows an ϵ-greedy policy: it either selects a random action with probability ϵ to allow exploration or chooses the action with the highest predicted Q-value to exploit current knowledge.

Once an action is taken, the environment returns the reward, the next state, and supporting information. Each transition is stored in the replay buffer, from which random mini-batches are sampled during training. For every sampled transition (s, a, r, s'), a target value is computed using the target network:

$$y = r + \gamma \max_{a'} Q(s', a'; \theta^-),$$

where γ discounts future rewards. The Q-network parameters θ are then updated by minimizing the squared difference between this target and the current estimate $Q(s, a; \theta)$. After every C step, the target network is synchronized with the online network to improve learning stability. On completing all training episodes, the optimized Q-network can be used to recommend medications by selecting actions that maximize predicted therapeutic benefit for each RLGMO state.

To assess the effectiveness of our trained reinforcement learning model, we analyzed its recommendations against real-world patient data and clinical references. Table 5.1 presents a summary of selected test cases highlighting the model's behavior across various patients, medications, and contextual similarities.

TABLE 5.1: Sample Results of Therapy Recommendation using RL-GMO Model

(A) Recommended Medications and Patient (RLMO) Context

RLGMO_ID	Suggested Med	Actual Medications	Reward	Match Type
XYZ035	leucovorin	ibuprofen, prednisone, chemotherapy, omeprazole	1.000000	similar_RLGMO
XYZ036	iohexol	bevacizumab, tamoxifen, letrozole, therapy	0.986425	similar_RLGMO
XYZ037	leucovorin	cyanocobalamin, benzonatate, mometasone	0.993805	similar_RLGMO
XYZ038	clotrimazole	oncoptype, calcium, docetaxel	0.969700	similar_RLGMO
XYZ039	pantoprazole	paclitaxel, pertuzumab, ondansetron	0.903594	similar_RLGMO
XYZ040	leucovorin	flexeril, ibuprofen, simethicone, tramadol	1.000000	partial_RLGMO

(B) Similarity Scores and Reference Treatment Details

Matched Terms	Cosine Sim.	Matched RLGMO_ID	Ref. ID	Ref. Treatment
Symptoms: fatigue, nausea	0.844894	XYZ016	168	Stress management and Relaxation techniques, Support group, Antidepressant
Symptoms: numbness	0.954751	XYZ013	–	–
Symptoms: nausea	0.979350	XYZ016	–	–
Disease: tumor	0.899000	XYZ018	12	The tumor may just require monitoring. When treatment is required, it may include radiation or surgical removal.
Disease: diabetes; Symptoms: nausea	0.345315	XYZ021	213	Artifical insulin(Type 1 diabetes), A diet change(Type 2 diabetes), Metformin(Type 2 diabetes).
Symptoms: fatigue, nausea	0.683805	XYZ016	168	Stress management and Relaxation techniques, Support group, Antidepressant

For training and validating the RLGMO model, a total of 40 patient records were employed. The patient records were split into 34 patients (RLGMOs), or 85%, for training purposes and 6 patients (RLGMOs), or 15% of the total records, for determining the generalization performance of the model. This patient split allowed the evaluation of how well the model was capable of suggesting suitable treatment plans for a patient's profile that had not been seen beforehand, which is a very essential task in a clinical environment.

Each row in Table 5.1 corresponds to a separate case in the test data. The column fields include the recommended treatment, the prescribed treatment, the reward value calculated using the reinforcement learning algorithm, the match type (such as similar_RLGMO, exact match, or guideline match), and the cosine similarity value between the GCN representation of the RLGMO and that of the matching existing case. The last two fields, Reference_ID and Reference_Treatment, denote the matching between the recommendation output and the existing guidelines.

High-Confidence Matches Patients XYZ035, XYZ037, and XYZ040 achieved rewards of 1.0 or nearby, indicating a strong alignment between the recommended

and actual medications. These cases also exhibited high cosine similarity scores (>0.9) and a similar_RLGMO match type, suggesting that the reinforcement learning policy effectively leveraged graph-based embeddings to identify semantically and clinically relevant past cases. This behavior validates the model's capacity to learn interpretable and transferable treatment representations, demonstrating that graph embeddings meaningfully capture RLGMO similarity in multi-dimensional medical contexts.

Guideline Validation RLGMO XYZ039 provides a solid example of external validation. The drug that the model predicts, pantoprazole, is in line with guideline_ID [213], which deals with managing diabetes and suggests a mix of dietary and insulin interventions. This shows that the approach improves clinical interpretability and guarantees consistency with recognized therapeutic standards by both recognizing direct RLGMO similarity and generalizing through reference guideline matching.

Reference and Equivalency Gaps Certain cases, such as XYZ036 and XYZ037, achieved high similarity and reward scores but lacked corresponding matches in the reference or RxNorm-equivalent medication fields due to unmatched medical terms. These reference failures indicate potential gaps either in the coverage of the guideline dataset or in the drug class equivalence resolution process used during model inference. Addressing these limitations by expanding reference mappings or improving RxNorm-based ontology matching would further increase the robustness and clinical validity of the RLGMO system.

Reward Dynamics and Semantic Contribution An interesting pattern is observed in XYZ039, where despite the lowest cosine similarity (0.345315) among high-reward cases, the RLGMO still achieved a relatively high reward (0.903594). This behavior demonstrates that the reward function captures not only vector-level embedding similarity but also semantic-level pattern alignment across symptoms, diseases, and medications. This type of dynamic reward formulation allows the model to acknowledge meaningful partial matches and contextual overlaps, enhancing flexibility when direct vector correlations are weak.

Overall Assessment and Validity Overall, Table 5.1 demonstrates the model's strong ability to generalize across diverse RLGMO cases by recognizing consistent therapeutic patterns, then integrating multiple layers of validation, and adapting effectively to complex or uncertain clinical situations. The following points summarize the major findings and their significance:

- **GCN embeddings:** The graph convolutional network effectively transforms each RLGMO's medical information—such as diseases, symptoms, and prescribed medications—into meaningful numerical representations. These embeddings preserve interconnections among clinical entities, enabling the model to group similar RLGMOs and uncover patterns that may not be directly visible in tabular data. This structured representation provides the foundation for downstream analyses such as RLGMO similarity search and therapy recommendation.
- **Reinforcement learning agent:** The reinforcement learning component optimizes treatment decisions through a reward-driven process, learning from

repeated feedback and performance outcomes. It dynamically refines its recommendations to align with successful historical cases and established treatment strategies. Over time, this agent demonstrates the ability to balance exploration of new therapy options with reliable adherence to clinically sound choices.

- **Guideline and RxNorm validation:** Since the dataset contains a limited range of medication examples, external validation through authoritative sources becomes essential. By incorporating medical guidelines and RxNorm-based mappings, the system can recognize alternative drug brands or equivalent formulations within the same therapeutic class. This integration not only compensates for data scarcity but also strengthens clinical reliability, ensuring that recommended therapies remain medically valid, standardized, and aligned with regulatory frameworks.
- **Reward-based adaptability:** The model maintains performance even when RLGMO similarities are weak or incomplete. Through its reward feedback mechanism, it adjusts to less common clinical profiles or edge-case scenarios without losing interpretability. This adaptability reflects the robustness of the model's learning strategy and supports its scalability to broader, real-world healthcare environments.

From a methodological perspective, these outcomes validate the reliability and interpretability of the RLGMO module. The strong agreement between its recommendations, historical clinical outcomes, and standardized medical guidelines demonstrates that the model not only learns from data-driven experiences but also integrates medically grounded reasoning. This dual mechanism—combining graph-based learning and reinforcement optimization—establishes a credible foundation for future deployment in clinical decision support systems, emphasizing transparency, reproducibility, and compliance with FAIR and ethical AI principles.

Answer to RQ$_3$. In this research study, a model-free reinforcement learning agent based on a Deep Q-Network (DQN) was used, utilizing GCN-derived RLGMO embeddings and a dual-reward mechanism to iteratively learn optimal therapy strategies for future incoming data. The dual-reward logic balances historical treatment effectiveness with clinical guideline validity, enabling the system to generate personalized and adaptive medication recommendations. This approach allows the model to dynamically optimize therapy decisions while maintaining interpretability and medical reliability.

5.5 Discussion

The results of the proposed reinforcement learning framework demonstrate the effectiveness of integrating graph-based patient representations with model-free decision-making algorithms for personalized therapy recommendations in oncology. Using GCN-derived embeddings in combination with a Deep Q-Network (DQN) agent trained on historical patient data and validated reference guidelines, the system successfully generated clinically meaningful medication suggestions across a range of oncology cases.

Our recommendation pattern analysis, shown in Table 5.1, highlights the strong model performance in cases such as XYZ035, XYZ037, and XYZ040, where high-confidence matches were obtained through symptom–disease similarity. These cases described the ability of the GCN module to capture latent semantic relationships within the knowledge graph and support the DQN agent in identifying therapeutically consistent treatment strategies. The combination of structural similarity and reward-driven learning reflects the framework's ability to merge relational reasoning with adaptive optimization.

Furthermore, the model's alignment with medical guidelines—such as the correct diabetes-related treatment identified for XYZ039—underscores its potential for clinical applicability. The integration of RxNorm-based drug equivalency mapping enhanced generalization by linking branded and generic medication variants, enabling more robust cross-patient reasoning.

Despite these strengths, certain limitations remain. For example, XYZ036 and XYZ037 demonstrated high RLGMO similarity yet lacked valid reference alignment or exact medication matches, primarily due to incomplete coverage in the reference dataset and limited drug-class mappings. Additionally, some cases, such as XYZ039, revealed that the model could still propose clinically acceptable medications despite low embedding similarity. This suggests that the dual-reward logic successfully captures semantic relationships beyond purely numerical similarity but may benefit from further balancing between symbolic reasoning and vector-based pattern recognition.

The performance of the RLGMO module is naturally dependent on the quality of the preceding components, such as entity extraction and knowledge graph construction. Inaccurate identification of entities, lack of clinical relations, or incorrect connections in the knowledge graph can affect the patient embeddings and, therefore, the reinforcement learning policy. Thus, errors accumulated in the preceding stages of the pipeline can affect the therapy recommendation process itself.

5.5.1 Implications

These findings further strengthen the second major component of the MetaGRO Framework, which focuses on graph-based reasoning and reinforcement learning for therapy optimization. By combining Graph Neural Networks (GNNs) with Reinforcement Learning (RL), this stage of MetaGRO provides a powerful, interpretable, and adaptive foundation for medical decision support. Unlike static or rule-based systems, the RLGMO module continuously learns from experience, enabling deeper pattern discovery, explainable graph-based inference, and dynamic adaptation as new patient data becomes available.

The clear alignment between the model's recommendations and established clinical guidelines, as well as its ability to recognize drug-class equivalence, highlights both reliability and clinical relevance—qualities essential for real-world deployment. The transparent structure of the patient knowledge graph and the reward-explanation mechanism also help enhance clinician trust, which is fully consistent with the FAIR and ethical-AI principles emphasized in the MetaGRO Framework.

Looking ahead, future work will extend the capabilities of the RLGMO component by incorporating temporal patient trajectories, scaling to multi-disease datasets, and integrating advanced RL paradigms such as Actor–Critic and Proximal Policy Optimization (PPO). Expanding medical reference sources and enabling real-time clinical feedback will further improve robustness, adaptability, and clinical readiness.

Overall, the results demonstrate that the second stage of the MetaGRO Framework successfully bridges graph-based knowledge representation, reinforcement-driven learning, and clinical interpretability—creating a strong foundation for personalized, data-driven therapy optimization within modern oncology care.

5.5.2 Threats to Validity

This section highlights the factors that may affect the reliability of the results from the knowledge-graph and reinforcement-learning experiments.

- **Internal validity.** Internal factors—such as possible bias in the baseline comparisons and the way the dataset was preprocessed—may influence the results. To reduce this risk, all methods were evaluated on the same dataset using identical hyperparameters and metrics, ensuring consistency and reproducibility.
- **External validity.** The generalizability of the framework may vary when applied to datasets from other hospitals or clinical settings that follow different formatting styles. Including more diverse, multi-institutional datasets in future work will help improve robustness and real-world applicability.
- **Conclusion validity.** The experimental conditions, including the limited test set and simulated environment, may not fully represent real clinical scenarios. To minimize this threat, we used systematic validation and applied the same evaluation procedures across all models to maintain fairness and reliability.
- **Data-dependency validity.** The reinforcement-learning model depends directly on the structured information produced earlier in the workflow. Any errors or missing details in these inputs may affect the knowledge-graph construction and, later, the learning of treatment policies. Improving the quality and coverage of structured inputs in future stages can help minimize this issue.

Overall, while these limitations need to be kept in mind, the consistent experimental design and fair comparison across models give confidence in the reliability of the results presented in this chapter.

5.5.3 Scope and Deployment Boundary of the RLGMO Model

The RLGMO model is intended to be a clinical decision-support model rather than an independent medical decision-making system. The goal of the RLGMO model is to process structured patient representations that are extracted from past oncology data and provide recommendations for therapy based on learned patterns of similarity and reinforcement learning optimization.

The RLGMO model is not intended to replace the expertise of medical professionals or to provide prescriptive or mandatory recommendations for therapy. All therapy recommendations produced by the RLGMO model are to be validated and interpreted by qualified medical professionals. The model processes retrospective and carefully curated datasets and has not been validated through prospective clinical trials.

Additional validation procedures, such as extensive multi-center evaluation, bias analysis, robustness testing under distribution shifts, ethical review, and regulatory compliance in line with medical device governance frameworks, would also be necessary for real-world clinical deployment.

The current work exhibits organized integration within a research framework and methodological viability. Interdisciplinary cooperation between AI researchers,

oncologists, regulatory agencies, and medical facilities would be necessary for clinical translation.

5.6 conclusion

Our proposed approach advances traditional knowledge graph methodologies by combining graph-structured patient data with pattern recognition and adaptive learning through Graph Neural Networks (GNNs) and Reinforcement Learning (RL). Rather than analyzing clinical entities in isolation, the RLGMO (Reinforcement Learning Guided Medicine Optimization) module captures the semantic relationships among diseases, symptoms, medications, and patient histories. This enables the system to uncover meaningful disease–symptom–treatment patterns that guide an RL agent in generating adaptive, personalized therapy recommendations informed by historical outcomes and validated clinical references.

By structuring oncology patient information into relational graphs and embedding them using Graph Convolutional Networks (GCNs), the framework is able to perform context-aware reasoning across multiple clinical attributes. These graph embeddings provide interpretable representations for the RL policy, ensuring that the agent recommends therapies that align with historical evidence while still remaining flexible to patient-specific variations. This demonstrates that unifying knowledge representation with sequential decision-making can lead to transparent, data-driven optimization in personalized medicine.

Future work will focus on integrating the proposed model into a real-time Clinical Decision Support System (CDSS). In such a system, clinicians would interact using multimodal inputs, including clinical notes, diagnostic reports, and treatment histories. Such an environment would support evidence-based decision-making by enabling accurate disease prediction, personalized drug selection, and therapy planning. Beyond oncology, we also envision extending the RLGMO module to multi-disease management, pharmaceutical research, and clinical trial optimization —contributing to the broader goal of AI-powered, FAIR-aligned, and ethically responsible precision healthcare.

Although the RLGMO module facilitates adaptive and personalized therapy optimization through graph-based reasoning and reinforcement learning techniques, the proper management and dissemination of these results must be conducted within the boundaries of existing data management practices. Thus, the structured knowledge graphs, embeddings, and recommendations developed through the MetaGRO framework must be managed, assessed, and disseminated to ensure transparency, interoperability, and reproducibility. For these reasons, the second part of chapter 6, of this thesis will change its focus from algorithmic optimization to data management practices that align with the FAIR data principles and Open Science movement, exploring how these AI-generated results can be managed within research infrastructures.

In all, the proposed MetaGRO model unifies the following key components: assured BERT-based biomedical entity extraction, knowledge graph-based structured reasoning, graph embedding using GCNs, reinforcement learning-based optimization of the therapy, and FAIR-based dissemination. In the proposed model, the clinical intelligence pipeline and the governance framework are not seen as two separate and parallel tracks; instead, they are viewed as two complementary and interdependent layers. In this regard, the analytical capability of the graph-based reinforcement learning approach is further enhanced with the structured data stewardship.

Part II

Open Science and FAIR-a tools evaluation

Chapter 6

Background on OSP and FAIR Principle

In this chapter, we recall the fundamental concepts of Open Science and the FAIR (Findable, Accessible, Interoperable, and Reusable) principles and describe a general process for evaluating and improving data compliance in research infrastructures. Since the application of FAIR principles depends strongly on the data type and domain, we present two representative examples that illustrate the diversity of FAIR implementation challenges. These examples highlight how different research contexts demand distinct evaluation strategies.

Open Science Platforms: This focuses on infrastructures such as SoBigData RI, which promote transparent data sharing, collaborative analysis, and reproducible research. It demonstrates how FAIR compliance enhances openness and supports large-scale scientific cooperation.

Biomedical Data Management. This case explores the specific challenges of applying FAIR principles to medical datasets, where privacy, metadata completeness, and interoperability with clinical standards play critical roles.

The remainder of this chapter is organized as follows: Section 6.1 presents key concepts and workflows related to open science and FAIR principles. Section 6.2 introduces the evaluation framework for assessing FAIR-a tools. Section 6.3 introduces the FAIR-a tools used in this evaluation framework. The related work on these FAIR-a tools has been discussed in Section 6.4. Finally, Section 6.5 discusses how these tools contribute to more transparent and reusable scientific data practices.

6.1 What is Open Science and OSPs!

6.1.1 The concept of Open Science

It is a global movement that promotes transparency, collaboration, and accessibility throughout the research lifecycle. It encourages the open sharing of data, software, methods, and publications, allowing scientific work to be verified, reused, and extended by others. By reducing barriers to access, Open Science[1] enhances reproducibility, fosters innovation, and strengthens the connection between science and society. Its core practices are often summarized in six mutually reinforcing components shown in Figure 6.1, that are: (i) open methodology, (ii) open source / open software, (iii) open educational resources, (iv) open data, (v) open peer review, and (vi) open access.

[1] https://en.wikipedia.org/wiki/Open_science

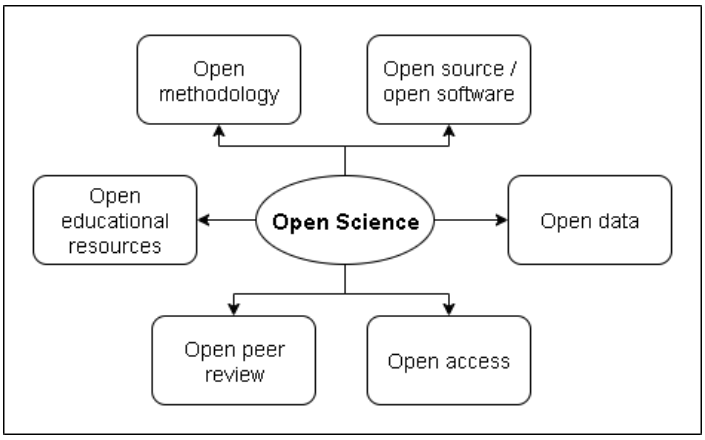


FIGURE 6.1: Representation of the open-science platform highlighting open access, data sharing, and collaboration.

6.1.2 Open Science Platforms

Open Science platforms (OSPs) facilitate data sharing, collaboration, and transparency among researchers worldwide. These platforms' integrated digital environments make it easy for researchers to save, exchange, and manage data and discoveries by ensuring that research is Findable, Accessible, Interoperable, and Reusable. These platforms, such as the European Open Science[2] Cloud (EOSC) and OpenAIRE, support interoperability between repositories and institutions, ensuring that scientific outputs are linked, properly cited, and openly accessible. Collectively, they represent a foundation for a more transparent, collaborative, and equitable scientific ecosystem. The assessment and implementation of FAIR principles have drawn sustained interest across disciplines, with evaluations spanning conceptual foundations, tooling, and practice. Since their broad adoption post-2016, FAIR has been credited with improving cross-disciplinary collaboration and data sharing through machine-actionable metadata, persistent identifiers, and standard vocabularies [136].

Several open science platforms support these principles by providing infrastructures for data and publication sharing. Notable examples include **Zenodo**[3], an open-access repository for datasets and research outputs developed under the European OpenAIRE program; **PubMed Central**[4], which offers free access to biomedical and life sciences literature; and **SoBigData**[5], a European research infrastructure for big data and computational social science. These platforms collectively exemplify FAIR principles and play a central role in enabling transparent, interoperable, and collaborative scientific research.

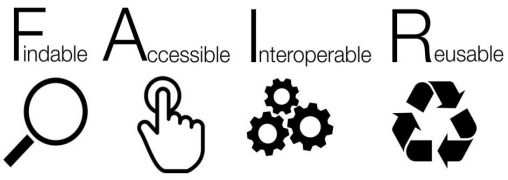


FIGURE 6.2: Representation of the FAIR Principles highlighting data Findability, Accessibility, Interoperability, and Reusability.

[2] https://research-and-innovation.ec.europa.eu/strategy/strategy-research-and-innovation/our-digital-future/open-science_en

[3] https://zenodo.org/

[4] https://www.ncbi.nlm.nih.gov/pmc/

[5] https://sobigdata.eu/

6.2 F.A.I.R Principles: The key concept

In 2016, the *FAIR Guiding Principles for scientific data management and stewardship* were introduced to improve the **F**indability, **A**ccessibility, **I**nteroperability, and **R**eusability of digital assets. The principles emphasize *machine-actionability*—the capacity of computational systems to find, access, interoperate, and reuse data with little or no human intervention—recognizing that researchers increasingly depend on computational support as data volume, complexity, and velocity grow. A concise "how-to" is offered by the Three-point FAIRification Framework.[6]

6.2.1 Findable.

The first step toward reuse is the ability to locate data. Machine-readable metadata are essential for automatic discovery.

- F1. (Meta)data are assigned a globally unique and persistent identifier.
- F2. Data are described with rich metadata (as further specified by R1).
- F3. Metadata clearly and explicitly include the identifier of the data they describe.
- F4. (Meta)data are registered or indexed in a searchable resource.

6.2.2 Accessible.

Once found, data must be retrievable via standardized protocols, with authentication/authorization where appropriate.

- A1. (Meta)data are retrievable by their identifier using a standardized communications protocol.
- A1.1 The protocol is open, free, and universally implementable.
- A1.2 The protocol allows for an authentication and authorization procedure, where necessary.
- A2. Metadata remain accessible even when the data are no longer available.

6.2.3 Interoperable.

Data should integrate with other datasets and workflows using shared languages and vocabularies.

- I1. (Meta)data use a formal, accessible, shared, and broadly applicable language for knowledge representation.
- I2. (Meta)data use vocabularies that themselves follow FAIR principles.
- I3. (Meta)data include qualified references to other (meta)data.

[6] `https://www.go-fair.org/fair-principles/`

6.2.4 Reusable.

Maximizing reuse requires rich description, clear licensing, and provenance.

- R1. (Meta)data are richly described with a plurality of accurate and relevant attributes.
- R1.1 (Meta)data are released with a clear and accessible usage license.
- R1.2 (Meta)data are associated with detailed provenance.
- R1.3 (Meta)data meet domain-relevant community standards.

The FAIR principles apply simultaneously to three interrelated entities: the *data* (digital objects), the *metadata* describing them, and the *infrastructure* that provides registration, indexing, and access. For example, F4 requires that both metadata and data be registered or indexed in a searchable resource, underscoring the infrastructure's role in long-term discoverability.

6.3 Classification and Description of FAIR-a Tools

6.3.1 Online Survey-Based Self-Assessment Tools (OSBSAT)

Online survey-based self-assessment tools enable users to manually evaluate the FAIRness of datasets by responding to structured questionnaires. These tools are primarily designed to promote awareness of FAIR principles, encourage self-reflection, and provide preliminary insights into data management practices. Below, we describe several prominent examples in this category.

FAIR Self-Assessment Tool: Developed by the Australian Research Data Commons (ARDC) [137], [138], the FAIR Self-Assessment Tool is a questionnaire-based platform designed for educational and informative use. It was inspired by the CSIRO 5-Star Data Rating Tool [139] and the FAIRdat tool from DANS [140]. The tool's web-based interface presents a single-page questionnaire, with questions distributed across the four FAIR principles—Findable, Accessible, Interoperable, and Reusable. Users respond through multiple-choice and true/false options. The tool is distributed as an HTML file (index.html), which can be hosted on a local or institutional server. It has been collaboratively developed by Kerry Levett, Keith Russell, Martin Schweitzer, Kathryn Unsworth, and Andrew White, with open community contributions supported through its `CONTRIBUTING.md` guidelines.

FAIR Data Self-Assessment Tool: This tool is an adaptation of the ARDC FAIR self-assessment platform, developed by Cross-Lateral Enterprises Pty Ltd. It follows the same structure but is intended purely for informational and self-evaluative use. The tool acknowledges that multiple interpretations of the FAIR principles exist and encourages users to view their results as reflective indicators rather than absolute scores. Its purpose is to stimulate discussion and self-assessment around potential improvements in data FAIRness. Although structurally identical to the ARDC version, its scoring algorithms differ slightly, which may result in varying outputs.

FAIR-Aware: Developed by the FAIRsFAIR project, FAIR-Aware aims to enhance researchers' understanding of the FAIR principles and their practical implementation. The tool is discipline-agnostic and can be applied at any research stage—ideally before data deposition. It is organized into four thematic sections aligned with the four FAIR concepts. Each section presents a series of questions accompanied by guidance notes, explanatory pop-ups, and suggested resources to help users grasp underlying FAIR concepts. Responses are rated on a five-point Likert scale (from "strongly agree" to "strongly disagree"). The tool summarizes the user's FAIR awareness through quantitative and qualitative feedback, showing levels of "awareness" and "willingness to comply." It also allows users to retake the assessment to measure improvement over time, making it both an educational and evaluative instrument.

FAIRdat: FAIRdat offers a simplified assessment process using a short questionnaire that produces FAIRness scores across the four principles. It assigns points, averages them across principles, and reports percentile rankings, particularly emphasizing reusability. Although it prompts users to provide a DOI, results are not intended as formal certification but as a self-assessment guide. The tool uniquely integrates a dataset review system, inspired by product review models such as those on Amazon, where datasets are rated by reusers. Each reviewed dataset receives a FAIR "star" rating (up to five stars) and comments on its usability and compliance with FAIR principles. The overarching goal of FAIRdat is to foster transparency, encourage feedback loops, and promote FAIRness benchmarking across repositories.

FAIRDataBR: FAIRDataBR, developed at the Federal University of Paraíba (UFPB, Brazil), automates FAIRness verification through an intuitive web interface. It provides a question-based evaluation aligned with the FAIR principles and displays visual feedback using pie charts representing overall and per-principle scores (each out of ten). The tool includes links to explanatory resources for users unfamiliar with FAIR concepts. However, usability limitations have been noted—such as the absence of navigation buttons to return to previous sections, slow evaluation workflows, and translation inconsistencies between Portuguese and English interfaces. While straightforward for experienced users, beginners may find it difficult to interpret without prior FAIR knowledge. Despite these challenges, FAIRDataBR's visual reporting enhances user engagement and offers a comprehensive view of FAIR compliance.

NFDI4Culture FAIR-Check: The NFDI4Culture FAIR-Check tool was designed to assess FAIRness in cultural heritage research data. It comprises a 25-question survey linked to the NFDI4Culture Helpdesk, which offers personalized assistance for specific queries. The questionnaire encourages researchers to reflect on their data practices and identify areas for improvement rather than providing a quantitative FAIR score. It also provides contextual guidance through references to the "Guideline for FAIR Cultural Studies Research Data Management." Results can be exported as a PDF summarizing responses, supporting documentation, and recommendations. While the tool promotes self-awareness and good data stewardship, it lacks automated scoring and improvement suggestions, making it more suitable for qualitative evaluation than quantitative benchmarking.

SATIFYD: SATIFYD is derived from the DANS self-assessment framework and focuses on improving dataset FAIRness before repository submission. The tool guides users through FAIR principle–aligned questions and provides actionable recommendations to enhance compliance. It is particularly recommended for use during data preparation or prior to data deposition, offering researchers a structured checklist for achieving higher FAIR standards.

6.3.2 Semi-Automated Types (SAT)

Semi-automated FAIR assessment tools combine manual user input with partial automation through integrated scripts, APIs, or web services. These tools assist researchers in evaluating the FAIRness of datasets or digital objects by automatically retrieving, validating, or calculating selected indicators while still requiring human interpretation for contextual accuracy. They strike a balance between the simplicity of self-assessment surveys and the efficiency of fully automated evaluators, making them suitable for both researchers and data stewards seeking intermediate control and feedback during FAIR evaluation.

FAIRshake: FAIRshake is a comprehensive toolkit designed to systematically evaluate the FAIRness of digital resources. It provides a structured environment for linking digital objects to specific metrics and rubrics, enabling consistent and reproducible FAIR assessments. The results are visualized through a distinctive grid of colored squares—ranging from blue (good) to red (unsatisfactory)—that represents the FAIR score of a resource. Gray squares indicate metrics for which no response was provided, ensuring visual transparency in incomplete assessments.

The FAIRshake framework comprises multiple components, including:

- a full-stack web application with a search engine interface, backend database, and RESTful API;
- a Chrome browser extension and a bookmarklet for submitting assessments directly from any webpage;
- FAIR analytics modules that generate project-level statistical summaries of evaluations.

Its REST API adheres to the FAIR principles and is documented using SmartAPI, Swagger/OpenAPI, and CoreAPI standards, ensuring interoperability and accessibility. The system dynamically adapts its grid visualization to support multiple rubrics composed of various metric sets, allowing resources to be evaluated using different community-defined criteria.

Compared to earlier FAIRness evaluation approaches, FAIRshake introduces several key advantages: a centralized database for managing users, projects, metrics, and assessments; an integrated user interface for conducting evaluations; and browser-based tools that streamline FAIRness checks directly within researchers' workflows. Collectively, these features make FAIRshake a powerful and user-friendly platform for scalable and transparent FAIR assessment.

6.3.3 Online Automated Types (OAT)

Online Automated Types (OAT) represent the most advanced class of FAIR assessment tools, designed to evaluate the FAIRness of digital resources with minimal

or no human intervention. These systems leverage APIs, web crawlers, and automated metric implementations to test datasets, repositories, and metadata structures directly against community-defined FAIR maturity indicators. Their primary advantage lies in speed, reproducibility, and scalability—allowing researchers and institutions to assess large collections of data objects within minutes. Typically, OAT tools generate quantitative FAIRness scores, visual dashboards, and downloadable reports, making them suitable for integration into research infrastructures and FAIR compliance workflows.

FAIR Enough: FAIR Enough is an automated FAIR assessment tool that performs evaluations in a matter of minutes using only a dataset's URL or DOI. It supports community-defined compliance checks and incorporates core universal maturity indicators. The tool enables rapid and stable evaluation execution—typically completing most analyses in under a minute—without requiring a commercial license.

FAIR Enough includes a library that allows new maturity indicators to be defined, published, and registered, and supports ORCID authentication for creating collections and authoring assessments. Much of its metric-testing codebase draws from the "F-UJI" project, while its architecture combines a React- and Material UI-based front end with a FastAPI and MongoDB back end. Users can complete evaluations anonymously by simply entering a resource URL, after which they are prompted to select one of several predefined FAIR maturity indicator collections:

1. FAIR Enough Metadata Maturity Indicators (fair-enough-metadata).
2. FAIR Evaluator Maturity Indicators.
3. FAIR Enough Data Maturity Indicators (fair-enough-data).
4. FAIR Maturity Indicators for Rare Diseases.

After selection, the evaluation executes automatically, producing a summary report and FAIR score based on the chosen indicators.

F-UJI: The F-UJI web service programmatically evaluates the FAIRness of research data objects based on a standardized set of 17 core metrics developed by the FAIRsFAIR project. Each metric is clearly defined, making the assessment transparent and reproducible. "F-UJI" stands for "FAIR Test," derived from the Malay word *uji* meaning "test." It provides a robust, machine-actionable framework that supports collaboration among repositories and data providers.

Metrics follow the FAIRsFAIR format (e.g., FsF-F1-01D), where each code identifies the principle and resource type (data or metadata). Except for certain accessibility and interoperability sub-metrics (A1.1, A1.2, I2), all FAIR principles are fully covered. The system outputs a comprehensive visualization of FAIRness levels, reporting both maturity level (0–3) and corresponding FAIRness score (0–1). Results are summarized through pie charts and bar graphs, enabling users to identify missing components and track improvement across versions.

FAIR-Checker: FAIR-Checker, developed by IFB (ELIXIR-FR), is an automated evaluation interface built in Python to simplify the implementation of FAIR metrics. It validates the FAIRness of web resources by combining semantic web technologies with established metadata standards. The system automatically extracts embedded semantic annotations from web pages and enriches them using linked knowledge

sources such as DataCite, OpenAIRE, and Wikidata. It then verifies class and attribute integrity using the Linked Open Vocabularies (LOV), Ontology Lookup Service (OLS), and BioPortal. The final step validates resources against Bioschemas community profiles. Funded by the French Institute for Bioinformatics (IFB), FAIR-Checker represents a powerful semantic-driven approach to ensuring machine-readability and interoperability in online datasets.

FOOPS!: FOOPS! is a specialized web-based tool designed to assess the FAIRness of ontologies and vocabularies. It performs 24 distinct checks distributed across the four FAIR dimensions, adapting FAIR principles for semantic artifacts. The service examines ontology URI persistence, content negotiation, interoperability through existing vocabularies, and reusability through metadata quality. FOOPS! distinguishes itself from earlier tools such as Vapour7 (focused on negotiation quality) and OOPS! (focused on ontology pitfalls) by offering a FAIR-oriented perspective. The output includes a comprehensive breakdown of compliance across findability, accessibility, interoperability, and reusability, offering users a detailed understanding of ontology FAIRness.

OpenAIRE Validator – FAIR Assessment: The OpenAIRE Validator provides automated testing of data source compatibility with OpenAIRE Guidelines and FAIR standards. Integrated within the OpenAIRE PROVIDE platform, it supports repositories, CRIS systems, and aggregators entering the European Open Science Cloud (EOSC) ecosystem. The tool performs four major steps: (1) data source validation, (2) registration and linking to OpenAIRE networks, (3) metadata enrichment via the OpenAIRE Broker, and (4) usage statistics generation through the UsageCounts service.

Results are visualized through bar and radar charts representing maturity indicators—7 for findable, 11 for accessible, 12 for interoperable, and 9 for reusable. Output tables further detail content-based validation results, including rule names, descriptions, and record-level statuses. These features make OpenAIRE Validator one of the most comprehensive automated FAIR assessment environments available.

FAIR EVA (Evaluator, Validator & Advisor): FAIR EVA is a web-based and API-driven tool developed under the European Open Science Cloud to assess digital object FAIRness across repositories. It supports persistent identifiers (DOI, Handle) and repository-based configurations, allowing domain customization through a plugin system. FAIR EVA integrates the RDA FAIR Data Maturity Indicators and offers visual reports using pie and bar charts with color-coded FAIRness levels. Distinct advantages include scalability for bulk evaluations, flexibility for diverse disciplines, and integration capabilities through semantics and ontologies. However, some users report occasional backend server issues and language localization challenges, particularly with the Spanish interface.

FAIR Evaluator: The FAIR Evaluator, developed using Ruby on Rails, offers an advanced automated evaluation environment with four major components: (1) a registry of Maturity Indicator Tests (MI Tests), (2) a registry of MI Collections, (3) a pipeline to execute test collections on a given resource, and (4) a registry of FAIRness evaluation results. It operates on Google Cloud infrastructure and supports user

identification through ORCID. Users can import new metrics, create community-defined test collections, and execute evaluations via a web interface. The FAIR Evaluator thus acts as a central repository and execution engine for maturity indicators, advancing transparency and reproducibility in FAIR evaluations.

HowFAIRIs: HowFAIRIs is a Python-based automated framework that evaluates the FAIR compliance of software repositories hosted on GitHub or GitLab. It checks adherence to the recommendations from fair-software.eu and generates visual badges representing compliance levels. The tool analyzes repositories based on five criteria: public availability with version control, license inclusion, citation metadata, registry entry, and the presence of a checklist. Each badge color (red, orange, yellow, green) indicates increasing levels of compliance, with green representing full adherence. Though efficient, HowFAIRIs may encounter issues when analyzing large batches (>100 repositories). Nonetheless, it remains an essential tool for assessing FAIRness in research software, promoting best practices in reproducibility and openness.

6.3.4 Offline Survey-Based Self-Assessment Types (OFSBSAT)

Offline Survey-Based Self-Assessment Tools (OFSBSAT) are primarily designed for institutional or individual evaluation of FAIR compliance without requiring an online interface. These tools typically consist of structured templates—such as editable PDFs, spreadsheets, or forms—that guide users through FAIR principle-based questions and scoring frameworks. They allow organizations and researchers to perform FAIR evaluations independently, even in environments with limited technical infrastructure or restricted internet connectivity. While less interactive than web-based systems, these tools emphasize reflection, awareness, and long-term planning toward FAIR data stewardship and organizational readiness.

Do I-PASS for FAIR: The "Do I-PASS for FAIR" framework extends the FAIR principles beyond datasets to institutional practices, enabling organizations to self-assess their level of FAIR maturity. Developed by the LCRDM (National Coordination Point Research Data Management), the tool addresses how research institutions can evolve into "FAIR enabling organizations." It provides two main components: **(1)** a formal definition of a FAIR-enabling research organization and **(2)** a self-assessment instrument available as an editable PDF template.

Users assess their organization's FAIR maturity across three progressive levels—beginner, intermediate, and advanced—by responding to structured questions. Based on the resulting scores, organizations can develop targeted roadmaps for improving their research data management (RDM) policies, technical infrastructure, and FAIR-aligned workflows. The framework serves as both a benchmarking and capacity-building tool, helping institutions translate FAIR principles into practical, organizational-level actions within Open Science initiatives.

FAIRness Self-Assessment Grids: The FAIRness Self-Assessment Grids, introduced by the Research Data Alliance's SHARC Interest Group, were designed to promote better recognition and credit for researchers who adopt FAIR data-sharing practices. These grids focus on the *human interpretability* of FAIR principles—complementing automated evaluations by emphasizing researcher comprehension and literacy.

Unlike tools that focus solely on machine-readability, the FAIRness Self-Assessment Grids encourage human-centered FAIR compliance through structured, reusable templates. Each grid provides decision-tree–like evaluation criteria, helping users prioritize relevant actions, support, and training based on their current level of compliance.

The assessment is available in an Excel format (version 1.1) under the publication *"Templates for FAIRness Evaluation Criteria – RDA-SHARC IG"* (June 29, 2020). Key features include:

- Comprehensive and rapid assessment using structured grids,
- A decision-tree design that supports stepwise FAIRification, and
- A strong focus on researchers and their practical engagement with FAIR data practices.

These grids provide an adaptable foundation for continuous FAIR assessment and improvement, empowering individuals and teams to evaluate and enhance their data-sharing practices over time.

6.3.5 Other Types (OT)

Other Types (OT) include frameworks and platforms that indirectly contribute to FAIR data implementation but are not primarily designed as assessment tools. These systems often focus on promoting good data stewardship practices, integrating FAIR principles into research workflows, and providing domain-specific training or guidance. Rather than scoring datasets against FAIR maturity indicators, they enhance FAIR compliance through planning, resource management, and knowledge dissemination. Such tools serve as essential complements to automated or self-assessment methods by embedding FAIR-by-design approaches within research and data management processes.

Data Stewardship Wizard: The Data Stewardship Wizard (DSW) aims to improve the perception of data management planning by emphasizing its benefits for researchers and research projects rather than its administrative burden. It helps users identify appropriate tools, metadata standards, and provenance documentation methods that align with FAIR principles—ensuring data is Findable, Accessible, Interoperable, and Reusable for both humans and machines.

Unlike most FAIR assessment tools that evaluate specific datasets, DSW assesses entire research projects. It features multiple-choice, open-ended, and unique identifier (GUID/DOI) questions to guide data management planning from project initiation. The Wizard can be used from the earliest stages, even before data creation, facilitating structured and transparent data organization throughout the project lifecycle. In addition to the four FAIR dimensions, DSW also evaluates two supplementary dimensions—data openness and data management. It provides actionable guidance for improving data handling during research, making it particularly suitable as a continuous planning and advisory framework rather than a rapid FAIR evaluation tool.

Triple Toolkit: The GoTriple platform, part of the OPERAS Research Infrastructure, represents an innovative multilingual discovery platform for the Social Sciences and Humanities (SSH). It provides one of the main access points for discovering and reusing research artifacts across diverse SSH disciplines. GoTriple automatically aggregates and semantically enriches publications, research data, project descriptions, and researcher profiles from various repositories and aggregators.

The associated Triple Toolkit extends these capabilities by offering practical materials to support FAIR-by-design training and community engagement. This includes planning templates for online training events, communication and outreach guidelines, and survey instruments to assess training needs and measure impact. All toolkit components are publicly available through Zenodo, promoting transparency, reproducibility, and FAIR adoption across SSH communities.

FAIRplus: The FAIRplus initiative, closely linked to the "FAIR Cookbook," seeks to advance the adoption of FAIR data principles within the life sciences. It introduces the FAIRplus-DSM (DataSet Maturity) model—an all-encompassing framework for progressively improving the FAIRness of research datasets. The FAIR-DSM Assessment Tool defines five maturity levels that guide datasets from basic metadata compliance to enterprise-level data governance:

- **Level 1:** Basic objects with simple metadata enabling fundamental discovery.
- **Level 2:** Enhanced structured data fields improving usability and data quality.
- **Level 3:** Conformance to community standards supporting advanced search and retrieval.
- **Level 4:** Cross-domain interoperability facilitating broader data integration.
- **Level 5:** Enterprise-level data management ensuring comprehensive governance and reliability.

The FAIRplus project provides a range of resources, standards, and "recipes" for practical FAIR data implementation tailored to diverse user groups, including researchers, data managers, and developers. Through its DSM model and Cookbook, FAIRplus bridges the gap between FAIR principles and real-world data management practices in biomedical research.

6.4 Related Work on FAIR-a Tools Assessment

The research is structured around a Systematic Literature Review (SLR), informed by three WH-questions that delineate the conceptual space of FAIR principles, FAIR-a tools, and their evaluation:

- **What** do the FAIR Principles represent, and how is "FAIRness" conceptualized in the literature?
- **Which** tools exist to support or assess the implementation of the FAIR Principles?
- **How** are these tools evaluated—through metrics, assessment methods, or FAIRness-oriented criteria?

These significant concerns guided the design of our search strategy and ensured that the review captured publications addressing the conceptual foundations of FAIR, the tools produced to operationalize FAIRness, and the methodology used to evaluate such tools. To operationalize this structure, we organized our search into three keyword groups:

- **FAIR** — "FAIR Principles", "FAIR Guiding Principles", "FAIRness", "FAIR Guidelines", "Data FAIRness".
- **TOOL** — "FAIR Tools", "FAIR Assessment Tools".
- **EVALUATION** — "FAIR Tools Evaluation", "FAIRness Tools Evaluation", "FAIR Metrics", "FAIR Assessment Metrics".

This curated corpus allows us to understand (i) How FAIR is interpreted as a set of guiding principles rather than a rigid specification, (ii) Which tools have emerged to operationalize FAIRness, and (iii) How such tools are assessed in terms of metrics, usability, and methodological robustness. These insights inform the methodological choices of our work, including the metadata fields considered, the provenance traces collected, and the evaluation criteria applied.

LISTING 6.1: Scopus advanced query used in this review.

```
TITLE-ABS ( ( "FAIR Principles" OR "FAIR Guiding Principles
   ↪ " OR "FAIRness" OR "FAIR Guidelines" OR "Data
   ↪ FAIRness" )
AND ( "FAIR Tools" OR "FAIR Assessment Tools" OR "FAIRness
   ↪ Tools Evaluation" OR "FAIR Tools Evaluation" OR "FAIR
   ↪  Metrics" OR "FAIR Assessment Metrics" ) ) AND
   ↪ PUBYEAR > 2015 AND PUBYEAR < 2025
```

The search query returned 30 papers; we excluded 12 that addressed algorithmic *fairness* rather than the data-stewardship FAIR paradigm, leaving 18 papers that constitute our core corpus (Table 6.1). Conceptually oriented works emphasize that FAIR is a set of guiding principles rather than a single specification and that practical value emerges when communities align on metadata richness, identifier practice, provenance capture, and standards adoption [136]. This framing informs our methodological choices (metadata fields, provenance traces, and evaluation criteria) and motivates instruments that make FAIR operational and comparable across domains.

TABLE 6.1: Summary of Related Work on FAIR Principles and FAIR-a Tools.

Reference or Citation	**Contribution**	**Research Gap**
Michaelis, Lea *(2023)[141]*	Evaluates the application of FAIR principles in the German Network University Medicine (NUM) during the COVID-19 pandemic using tailored assessment tools.	Highlights the lack of comprehensive evaluations of FAIR compliance and best practices in large-scale research networks such as NUM.
Trifan, Alina *(2020)[142]*	Provides an overview of the impact of FAIR principles and evaluates biomedical platforms using FAIR metrics.	Identifies challenges in translating FAIR principles into practice, particularly in assessing FAIR compatibility in biomedical platforms.
Trifan, Alina *(2019)[143]*	Evaluates FAIR adoption in biomedical platforms using FAIR metrics.	Notes limited examples demonstrating practical translation of FAIR principles in biomedical platforms.
Cuno, Alvaro *(2020)[144]*	Evaluates five stress detection datasets using FAIR metrics and identifies compliance gaps, especially in interoperability.	Shows lack of interoperability and FAIRness improvements in stress detection datasets.
Bahlo, Christiane *(2024)[145]*	Developed the AgReFed FAIR tool to enhance FAIRness in agricultural research data.	Existing FAIR tools lack functionality tailored specifically to agricultural research needs.
Gao, Ruoyuan *(2022)[146]*	Proposes a FAIR metric and algorithm for optimizing user utility and fairness in ranking systems.	Existing metrics inadequately combine fairness and user utility in search and recommendation systems.
Jones, Sarah *(2020)[147]*	Reviews data management planning tools and discusses their contributions to FAIR support.	Limited integration of FAIR principles into DMPs and issues with user competency in applying them.
Candela, Leonardo *(2024)[148]*	Analyzes 20 FAIR tools and 1,180 metrics, identifying trends, gaps, and inconsistencies in FAIR assessments.	Challenges persist in aligning metric intent with FAIR principles and addressing recurring issues in assessment tools.
Weber, Tobias *(2018)[149]*	Proposes a use-case-centric metric for assessing FAIR compliance in research data repositories.	Lack of specialized metrics for specific use cases such as spatially or temporally annotated images.
Clarke, Daniel J.B. *(2019)[150]*	Developed the FAIRshake toolkit for community-driven FAIR metrics and assessment of biomedical digital resources.	Few tools integrate manual and automated FAIR assessment methods with visualization capabilities.
Gaignard, Alban *(2023)[151]*	Developed FAIR-Checker for automated FAIRness assessment and metadata improvement.	Automation requires technical expertise, limiting accessibility for non-technical users.
Shiferaw, Kirubel Biruk *(2024)[152]*	Introduces a framework aligning reporting guidelines with FAIR principles to improve reproducibility and open science.	Limited integration between FAIR principles and traditional reporting guidelines limits reproducibility.
Craig, Adam *(2019)[153]*	Introduces the DREAM principles as a historical precursor to FAIR and defines quantitative evaluation metrics.	Raises concerns about originality, terminology consistency, and attribution within FAIR principles and metrics.
Wilkinson, Mark D. *(2023)[154]*	Defines FAIR stakeholder categories and explores governance models for FAIR implementation.	Lack of standardized measurement of FAIR behaviours across stakeholders.

Continued on next page

Reference or Citation	Contribution	Research Gap
Müller, Heimo *(2023)[155]*	Developed BIBBOX, a platform that enables FAIR-compliant publication of datasets and software.	Ambiguity in FAIR implementation impedes effective data sharing in life sciences.
Lien-Talks, Alphaeus *(2024)[156]*	Highlights the importance of FAIR principles in bioarchaeology and proposes standardized data management and federated search interfaces.	Lack of standardized procedures and metadata documentation affects effective data reuse in bioarchaeology.
Feijoó, Matheus Pedra Puime *(2020)[157]*	Introduces Bio FAIR Evaluator to assess genomic database FAIRness from both human and machine perspectives.	Existing work lacks detailed dual-perspective evaluations of genomic database FAIRness.
Krans, N.A. *(2022)[158]*	Reviews ten FAIR assessment tools, categorizes their usability, and provides recommendations for users and developers.	Inconsistent implementation of FAIR principles across tools leads to varied assessment outcomes.
Thompson, Mark *(2020)[136]*	Explores emerging FAIR tools and technologies in their early creolization phase.	Immature FAIR tools and a lack of standardized practices hinder efficient adoption of FAIR principles.

A substantial body of work seeks to operationalize and measure FAIR through tools, checklists, and platform services. Community toolkits such as *FAIRshake* combine automated and manual assessments to score digital resources [150]. National or repository-level initiatives provide self-assessment forms and guidance, together with specifications for evidencing indicators in practice [159], [160]. Comparative evaluations consistently report variation in scope, metric definitions, evidence requirements, and automation level, which can yield divergent scores for the same resource [158], [159].

Domain deployments surface additional constraints. Michaelis et al. evaluated COVID-19 data within the German Network University Medicine and identified practical gaps and frictions in implementation [141]. Trifan et al. examined discoverability and reproducibility in biomedical contexts via FAIR-aligned audits [142], [143]. Sector-specific studies extend this lens to information retrieval and agriculture, showing how indicator weighting and evidence granularity affect outcomes [145], [146]. Tool-focused analyses document technical and usability pain points that hinder adoption or reproducibility [148], [149]. Recent efforts emphasize interoperability across repositories and genomic resources, raising issues around schema alignment and machine-actionability at scale [155], [157].

Synthesis and identified gaps. Across studies, three gaps recur. *First*, indicator definitions and their operationalization differ across tools, producing non-trivial score dispersion and complicating cross-study comparison [158], [159]. *Second*, runtime and user-experience factors (e.g., latency, stability, and evidence-capture burden) are often under-specified despite their importance for real-world uptake [148], [149]. *Third*, actionable guidance is limited: few tools translate results into prioritized, concrete remediation steps [158].

Positioning of this thesis. To address these gaps, we conduct a broad evaluation of **22 FAIR-a tools**—to our knowledge, among the most comprehensive to date—using a framework that covers four axes: *functionality*, *technical aspects*, *runtime behaviour*, and *usability and user satisfaction*. Beyond ranking, we report recurrent failure modes

(e.g., runtime inefficiencies, ambiguous metric wording, limited machine actionability) and provide concrete recommendations intended to (i) reduce effort during tool selection and (ii) guide developers toward changes that most improve FAIR outcomes.

Based on these outcomes, this part of the thesis outlines a unified methodology that addresses usability, technical performance, functionality, and runtime factors to thoroughly evaluate 22 well-known FAIR assessment tools, including open-source platforms like FAIRshake, F-UJI, and FAIR Evaluator. The assessment goal is to present one of the most robust comparison evaluations so far, pointing out frequent shortcomings and promising areas for development. Furthermore, as part of the continuous effort to expand the study's scope and improve the findings' generalizability, new FAIR tools are being evaluated.

6.5 Conclusion

Here this study provides a comprehensive overview and classification of existing FAIR assessment tools, categorized into four primary types—Offline Survey-Based Self-Assessment Tools (OFSBSAT), Online Survey-Based Self-Assessment Tools (OSBSAT), Semi-Automated Types (SAT), and Online Automated Types (OAT)—along with a group of supplementary tools under Other Types (OT). Each category reflects a distinct level of automation, technical complexity, and user interaction, allowing for a structured comparison of how these tools operationalize the FAIR principles across diverse research contexts.

The detailed analysis highlights that while offline and online self-assessment tools are instrumental in promoting awareness and organizational self-evaluation, automated and semi-automated systems offer scalability, objectivity, and machine-actionable FAIR-a tools assessments. Furthermore, the "Other Types" tools, such as Data Stewardship Wizard, Triple Toolkit, and FAIRplus, extend FAIR implementation beyond assessment, embedding it directly into research planning, training, and life-cycle data management. Together, these classifications reveal the evolving ecosystem of FAIR-a tools, where complementary approaches contribute to both human understanding and machine-based FAIR compliance.

Overall, this categorization into groups not only clarifies the functional scope and purpose of different FAIR-a tools but also identifies existing gaps and future opportunities. This bridging of usability and automation remains a key challenge—where integrating human-centered evaluation with computational precision can drive greater consistency and reliability in FAIR data assessment. The insights derived from this work fill the gaps in related work of other studies on FAIR-a tools evaluation.

Chapter 7

Insights and Improvement Approaches for OSPs

7.1 F.A.I.R Principles: An In-Depth Explanation

7.1.1 F – Findable the first key Principle

Definition: Findability refers to how easily desired information can be located within a digital platform or interface, particularly when handling information-rich resources. The term emphasizes the user's ability to discover information both within and outside a website or repository. Although the concept extends beyond the web, it was most notably popularized by Peter Morville [161] in 2005, who described findability as "the ability of users to identify a suitable website and navigate its pages to discover and retrieve relevant information resources."

In the context of FAIR data management, findability represents the first step toward effective data reuse. Both metadata and data should be easily discoverable by humans and machines. Machine-readable metadata are particularly crucial for automated dataset discovery and are therefore an essential component of the FAIR-ification process.

F1. (Meta)data are assigned a globally unique and persistent identifier. Globally unique and persistent identifiers (PIDs) remove ambiguity by assigning a distinctive reference to every dataset, metadata element, or measurement. This enables machines to interpret data meaningfully and ensures that humans can unambiguously identify and credit data sources. To meet this requirement:

1. **Global uniqueness:** The identifier must be universally unique, preventing duplication or reassignment. Registry services that employ strict uniqueness protocols (e.g., DOI, Handle, ARK) are used for this purpose.
2. **Persistence:** The identifier should remain stable over time. Since internet links often break or expire, registry services help ensure the long-term validity and accessibility of identifiers.

F2. Data are described with rich metadata (defined by R1). Even in the absence of the data itself, comprehensive metadata should enable the discovery of datasets. Rich metadata improves data visibility, increases citation potential, and enhances opportunities for reuse.

F3. Metadata clearly and explicitly include the identifier of the data they describe. This principle ensures an explicit link between metadata and their associated datasets. Because metadata and datasets are often stored as separate files,

including the dataset's persistent identifier within the metadata guarantees an unambiguous relationship between them.

F4. (Meta)data are registered or indexed in a searchable resource. Identifiers and metadata alone are not sufficient for findability. If a dataset is not registered or indexed in a searchable repository, it remains effectively invisible to potential users or automated systems. Repositories and indexing services implement principles F1–F3 to enable fine-grained discovery and ensure datasets can be found and accessed across domains.

7.1.2 A – Accessible: The second key Principle

Definition: Once data are located, they must be retrievable through standardized communication protocols. Accessibility ensures that both humans and machines can access metadata and data when needed, while allowing for authentication or authorization when appropriate. This principle focuses on the technical mechanisms that enable reliable access, promoting openness where possible and secure restriction where necessary.

A1. (Meta)data are retrievable by their identifier using a standardized communication protocol. Datasets and metadata should be accessible through universally accepted and documented communication protocols such as HTTP, FTP, or OAI-PMH. These standardized mechanisms ensure that any user or system can retrieve the information linked to a persistent identifier without ambiguity or dependency on proprietary systems.

A1.1 The protocol is open, free, and universally implementable. To guarantee long-term accessibility, the protocol used for data retrieval should be openly available, non-proprietary, and widely implementable. This promotes transparency and ensures that data can be accessed across different technological environments without licensing or financial barriers.

A1.2 The protocol allows for an authentication and authorization procedure, where necessary. While openness is a core value, certain datasets may contain sensitive or confidential information. In such cases, the retrieval protocol should support secure authentication and authorization workflows, allowing controlled access to authorized users while maintaining compliance with ethical and legal standards.

A2. Metadata remain accessible even when the data are no longer available. Even if datasets are deleted, restricted, or otherwise inaccessible, their metadata should remain publicly available. Persistent metadata ensures continued visibility, supports citation, and maintains the integrity of the scholarly record by documenting the existence, provenance, and context of the original data.

7.1.3 I – Interoperable the third key Principle

Definition: Interoperability ensures that data and metadata can be seamlessly integrated with other datasets, tools, and workflows. It emphasizes the use of shared languages, ontologies, and controlled vocabularies to facilitate automated data exchange and interpretation. The ultimate goal is to enable systems, both human and

machine, to understand, process, and reuse data across disciplinary and institutional boundaries.

I1. (Meta)data use a formal, accessible, shared, and broadly applicable language for knowledge representation. To ensure that data can be effectively interpreted by different systems, both metadata and data should use well-defined, standardized, and machine-readable formats. These formats—such as RDF, XML, or JSON-LD—enable semantic interoperability and allow diverse platforms to exchange and interpret information without loss of meaning or structure.

I2. (Meta)data use vocabularies that themselves follow FAIR principles. The vocabularies and ontologies used to describe data must also be FAIR: findable, accessible, interoperable, and reusable. By using established community-driven vocabularies—such as those in the Open Biomedical Ontologies (OBO) or Linked Open Vocabularies (LOV)—researchers ensure consistent terminology and facilitate cross-domain understanding. This approach prevents ambiguity and supports interoperability between datasets from different disciplines or repositories.

I3. (Meta)data include qualified references to other (meta)data. Datasets rarely exist in isolation. To enable contextual understanding, metadata should include qualified and explicitly defined references to related datasets, publications, or digital resources. These relationships provide valuable contextual links—such as derivations, citations, or dependencies—allowing users and machines to trace data provenance, verify findings, and perform integrated analyses.

7.1.4 R – Reusable the fourth key Principle

Definition: Reusability represents the ultimate objective of the FAIR framework, ensuring that data can be reliably used in future research contexts. To maximize reuse, datasets must be accompanied by comprehensive descriptions, clear licensing terms, and detailed provenance information. Well-documented data enable validation, reproducibility, and extension of scientific findings, supporting the long-term sustainability of research outputs.

R1. (Meta)data are richly described with a plurality of accurate and relevant attributes. High-quality metadata are critical for understanding and reusing datasets. Metadata should contain sufficient contextual information—such as experimental conditions, methods, instruments, and temporal coverage—to allow others to interpret and reproduce the data correctly. Rich, accurate descriptions enhance discoverability and enable meaningful comparison and integration with other datasets.

R1.1 (Meta)data are released with a clear and accessible usage license. Licensing defines the conditions under which data can be reused, shared, or modified. A clear, machine-readable license (e.g., Creative Commons, Open Data Commons) ensures that users understand their rights and obligations. Explicit licensing promotes transparency, prevents misuse, and encourages responsible data sharing across disciplines and organizations.

R1.2 (Meta)data are associated with detailed provenance. Provenance refers to the origin and history of data, including how, when, and by whom it was generated or modified. Recording provenance is essential for assessing data quality, reliability, and trustworthiness. It also facilitates reproducibility, as future users can trace how datasets were derived or combined during analysis.

R1.3 (Meta)data meet domain-relevant community standards. To promote reuse across scientific domains, data and metadata should comply with recognized community standards and schemas. Using standardized formats—such as MIAME for microarray data or DICOM for medical imaging—ensures compatibility and interpretability across platforms, tools, and research communities.

7.2 Comparison between open science repositories based on FAIRness

Before conducting the FAIR-based comparison, it is essential to briefly introduce the selected Open Science Platforms (OSPs) — SoBigData, Zenodo, and NCBI — as they represent diverse data management environments used widely in research communities.

SoBigData [162] is a European research infrastructure designed for big data analytics and open science collaboration. It provides access to social mining tools, datasets, and services supporting transparent, FAIR-compliant research.

Zenodo [163], maintained by CERN, serves as a multidisciplinary repository that enables researchers to deposit datasets, software, and publications under open licenses. It is a key platform for promoting accessibility and reusability across disciplines through DOI-based persistent identifiers.

NCBI [164] (National Center for Biotechnology Information) represents a large-scale biomedical data repository that integrates genomic, proteomic, and literature resources. It adheres to FAIR principles through standardized identifiers, cross-referenced metadata, and open programmatic interfaces.

These three platforms were selected because they differ in scope (social, general scientific, and biomedical), providing a comprehensive view of how FAIR principles are implemented across various domains. The following table summarizes their adherence to the core FAIR principles: Findability, Accessibility, Interoperability, and Reusability.

These three repositories were selected for analysis as they differ in scope—covering social science (SoBigData), general scientific (Zenodo), and biomedical (NCBI) domains—thus providing a comprehensive overview of how the FAIR principles are implemented across distinct research areas. In addition to these, other well-known FAIR-aligned repositories such as Figshare[1], Dryad[2], Harvard Dataverse[3], and EUDAT B2SHARE[4] also support open data sharing and reproducibility in various disciplines. The following table summarizes their adherence to the four core FAIR principles: Findability, Accessibility, Interoperability, and Reusability.

[1] `https://figshare.com`
[2] `https://datadryad.org`
[3] `https://dataverse.harvard.edu`
[4] `https://b2share.eudat.eu`

TABLE 7.1: Comparison of Open Science Platforms (SoBigData, Zenodo, and NCBI) Based on FAIR Principles

FAIR Principle	Open Science Platform (OSP)	Comparison Summary
Findable	SoBigData	Persistent identifiers (PIDs) are URLs. About 70% of articles are private, while the rest are public.
	Zenodo	DOIs serve as PIDs. Nearly all datasets and publications are public, with very few private items.
	NCBI	Persistent identifiers include DOI, OMID, and PMCID. Most articles are public, with few restricted.
Accessible	SoBigData	Metadata is accessible; however, most articles and datasets require authorization before login for full access.
	Zenodo	Both metadata and data are openly accessible via DOI. No login is required for viewing or downloading.
	NCBI	Data and metadata are accessible via PMID or DOI. Users can freely access and download most datasets.
Interoperable	SoBigData	English and Italian are used. References unavailable for private data but present for public datasets.
	Zenodo	English is used exclusively. All papers include structured references supporting interoperability.
	NCBI	English is the primary language. Dataset references are available via PubMed and PMC portals.
Reusable	SoBigData	Articles are registered but require multiple steps for reuse when private. Materials maintain provenance.
	Zenodo	Registered under MDPI licenses ensuring clear usage terms and provenance for all datasets.
	NCBI	Papers registered with PMCID and PMID numbers; associated materials show clear provenance.

7.3 Conceptual Idea: Using a Decision Tree to Improve FAIR Principles in the SoBigData.it Platform

The concept of the FAIR principles was first introduced in 2016. Since then, various techniques and software tools have been employed to enhance the FAIRness of data. The FAIR quality of data refers to the extent to which independent researchers can reuse the same dataset for experimentation. This aspect is crucial yet often underexplored. To address this gap, we propose a decision tree framework designed to guide researchers in assessing the findability, accessibility, interoperability, and reusability aspects of their dataset.

The decision tree assists researchers in determining whether a dataset complies with FAIR principles and whether additional metadata are required. It will also serve as the foundation for a future automated application that enforces FAIR standards by generating and attaching the necessary metadata according to the dataset's context (e.g., findability, accessibility, interoperability, and reusability).

Description: A decision tree is a supervised machine learning model used for classification or prediction, based on a sequence of conditional decisions. Each internal node represents a test on an attribute, each branch corresponds to a test outcome,

and each leaf node represents a final decision or classification label. The paths from the root to the leaves represent the rules applied for the classification.

In this work, the decision tree is structured to evaluate datasets in accordance with FAIR principles and to enhance their overall quality. The proposed question-based decision tree will appear on the repository's submission interface, where authors or researchers uploading sensitive or non-sensitive datasets can interact within it as a checklist to verify compliance with FAIR standards.

For the Findable principle, at the root level node, the tree first determines whether the dataset is public or private. Subsequent decisions and branches depend on this classification. If the dataset is private, the tree transitions to an intermediate phase, focusing on the methods used to access the data. If the dataset is public, it verifies whether all dataset instances are properly shared and examines FAIR-related aspects such as persistent identifiers, authoritative links, semantic validation, and data cleanliness or labeling. These checks collectively ensure that both public and private datasets comply with FAIR guidelines, thereby improving the discoverability, accessibility, and long-term usability of research data within the SoBigData platform.

7.3.1 Findability

The decision tree corresponding to the first FAIR principle, Findability, is presented below, along with its algorithmic representation.

Algorithm 12 outlines the procedure used to evaluate whether a dataset satisfies the FAIR *Findable* criteria. The algorithm checks the key conditions that make a dataset discoverable, beginning with its availability: whether the dataset is shared, whether alternative access routes exist, and whether clear instructions for accessing the data are provided. It then verifies the completeness and quality of the metadata, the presence of persistent identifiers, authoritative metadata links, and adherence to identifier schemas. At each decision point, if a required element is missing, the algorithm records the corresponding corrective action and returns an ActionRequired status. Only when all required conditions are fulfilled does the algorithm classify the dataset as Compliant, ensuring alignment with FAIR findability principles.

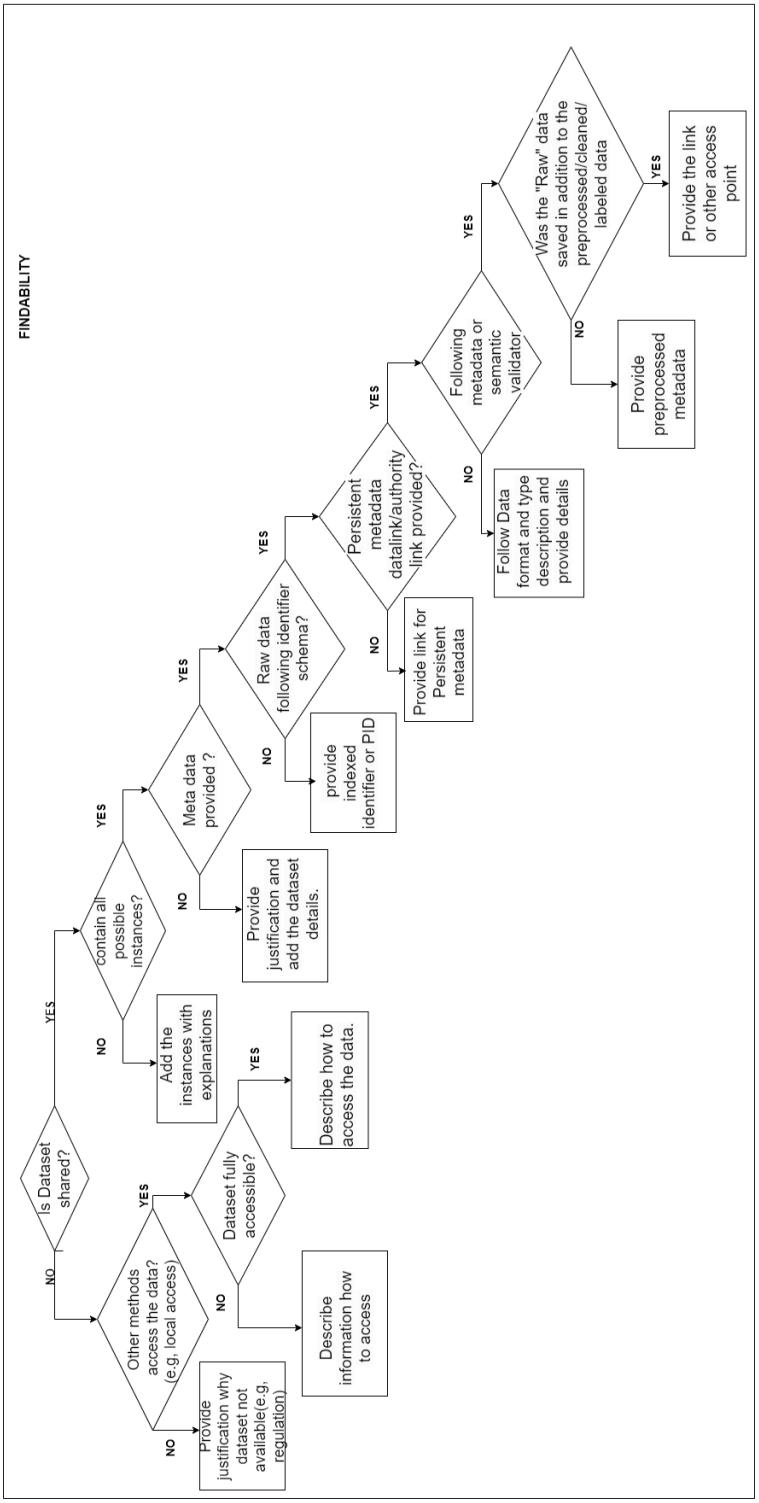


FIGURE 7.1: Decision Tree showing the Findability principle

7.3.2 Accessibility

The decision tree corresponding to the second FAIR principle, Accessibility, is presented below, together with its algorithmic representation.

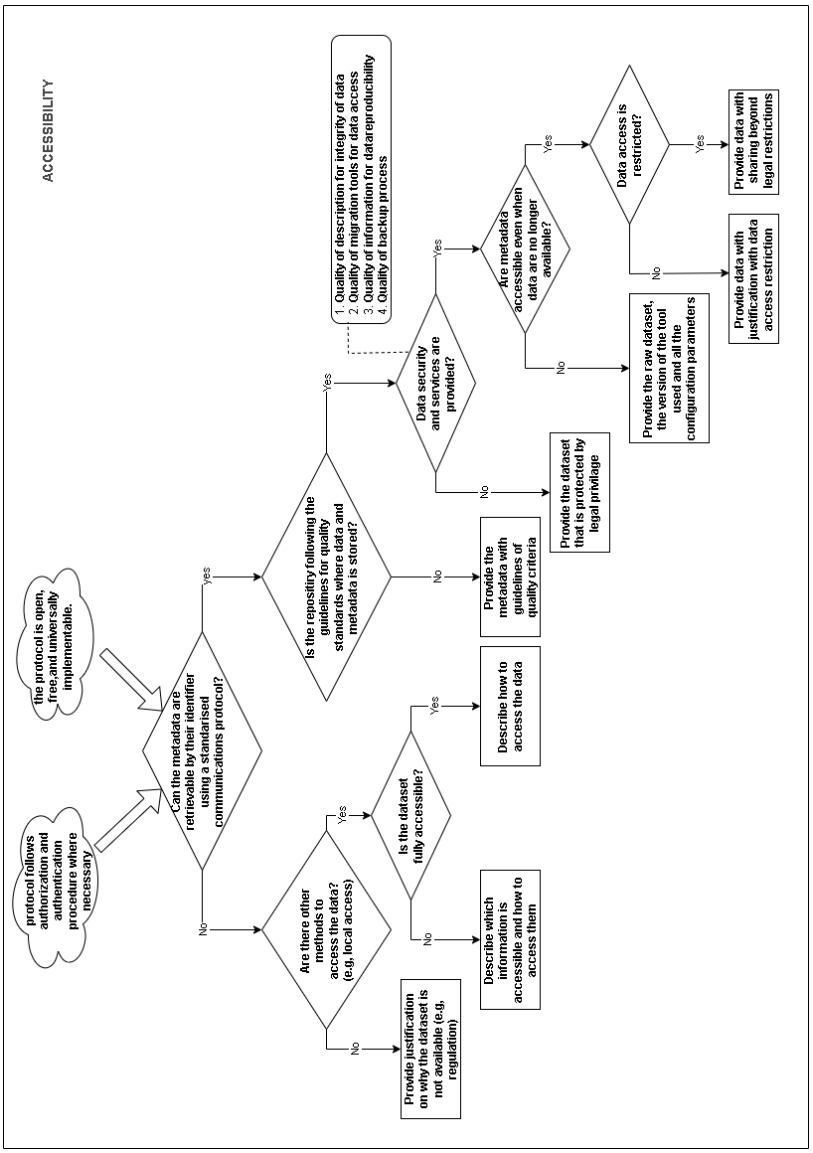


FIGURE 7.2: Decision Tree showing the Accessibility principle

The Algorithm 13 evaluates whether a dataset satisfies the FAIR *Accessibility* criteria. The procedure checks whether the dataset can be accessed through open, free, and universally implementable communication protocols and whether appropriate authentication or authorization mechanisms are in place when required. It then verifies that the metadata can be retrieved using a standardized protocol via the dataset's identifier. If no access route exists, or if the available access method is unclear or incomplete, the algorithm records the necessary corrective actions and returns an ActionRequired status.

It also examines whether the hosting repository follows recognized quality guidelines, whether adequate security measures and services are provided, and whether metadata remain accessible even if the dataset becomes unavailable. Finally, it assesses whether any access restrictions are justified or whether the dataset can be shared more openly. If all accessibility requirements are met, the dataset is marked as Compliant; otherwise, the algorithm outputs the list of actions needed to achieve FAIR-aligned accessibility.

7.3.3 Interoperability

The decision tree corresponding to the third FAIR principle, Interoperability, is presented below, along with its algorithmic representation. The Algorithm 14 evaluates whether a dataset satisfies the FAIR *Interoperability* criteria. The procedure begins by checking whether the dataset is interoperable—that is, whether it can be integrated with other datasets or tools without technical conflicts. If the dataset fails this basic requirement, the algorithm immediately records a justification request and returns an ActionRequired status.

Next, Algorithm 14 assesses the metadata to ensure that they use a formal, shared, and broadly applicable representation language. It also checks for the presence of standard vocabularies, thesauri, ontologies, or data dictionaries, which are essential for semantic consistency and machine interpretability. If these elements are missing, the algorithm identifies the required corrective steps.

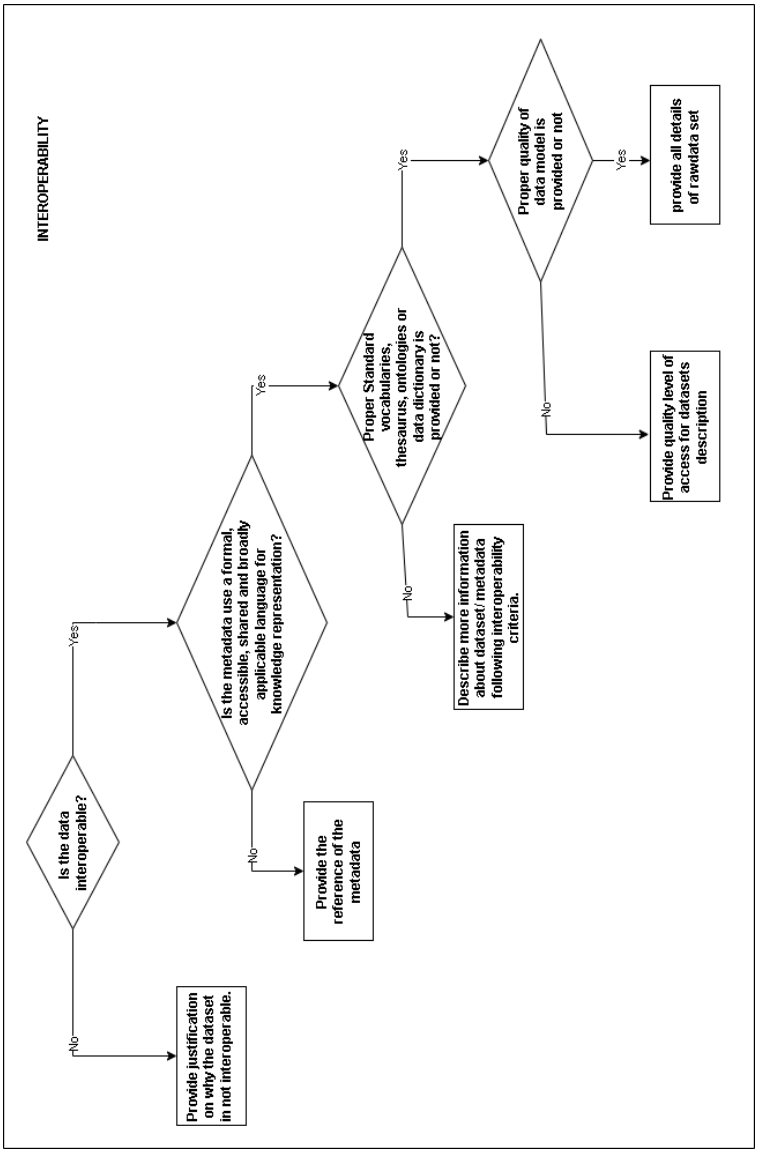


FIGURE 7.3: Decision Tree showing the Interoperability principle

The step further verifies whether an appropriate and well-defined data model is

provided to support the dataset's structural and semantic clarity. If all interoperability conditions are satisfied, the dataset is classified as Compliant, and the algorithm adds a final action note indicating that all raw dataset details should be supplied to complete the documentation.

7.3.4 Reusability

The decision tree corresponding to the fourth FAIR principle, Reusability, is presented below, together with its algorithmic representation.

Algorithm 15 assesses whether a dataset meets the FAIR *Reusability* criteria by examining the conditions that enable others to reliably use, reinterpret, or build upon the data. The procedure begins by evaluating whether the dataset can be shared. If the dataset cannot be made available and no alternative access route is provided, the algorithm requests a justification. If partial or controlled access exists, it records the necessary clarifications regarding what can be accessed and how.

This algorithm then inspects the dataset's legal and licensing conditions, ensuring that terms of use, copyright information, or reuse permissions are clearly stated. It also checks for explicit information describing the dataset's reuse potential, including provenance of derived data, curation protocols, and traceability of transformations. Further checks verify whether the dataset adheres to quality and enrichment practices, whether long-term preservation tools or services are available, and whether ethical and data-protection requirements for reuse are satisfied.

Each missing element results in an ActionRequired status along with the specific improvements needed. When all requirements for transparent licensing, provenance, traceability, quality assurance, and ethical reuse are fulfilled, the dataset is marked as Compliant, confirming alignment with FAIR reusability principles.

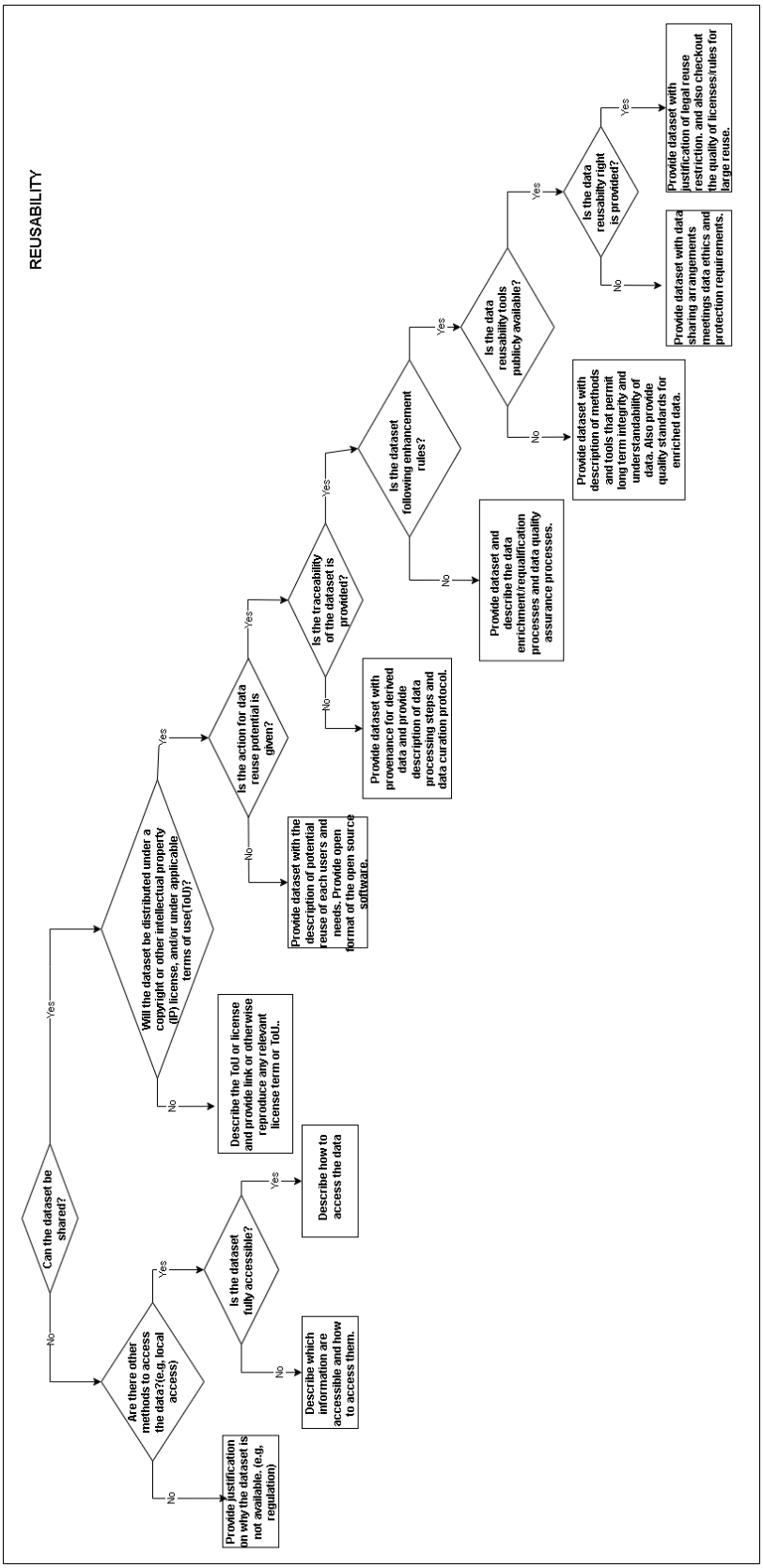


FIGURE 7.4: Decision Tree showing the Reusability principle

7.4 Improving SoBigData platform's FAIR principles through web-based services

7.4.1 FAIR Service Workflow

Figure 7.5 displays the detailed workflow concept of the FAIR Service integrated within the SoBigData platform. This service aims to enhance the FAIR compliance of datasets uploaded by users by guiding them through structured evaluation and feedback mechanisms. The workflow ensures that every dataset submission is examined against the FAIR criteria—Findability, Accessibility, Interoperability, and Reusability—before being stored in the repository.

The key steps of the workflow are summarized below:

- A registered user logs into the SoBigData platform and initiates a request to upload a new dataset.
- The platform automatically activates the **FAIR Service**, which launches the application form module.
- The application form operates based on a **decision tree** model as defined in [165].
- The decision tree interactively asks questions regarding the dataset's quality, structure, and metadata to ensure alignment with FAIR standards.
- This guided process helps the user progressively enhance the dataset's FAIRness by addressing potential gaps in compliance.
- Once the form is completed, the service performs the following actions:
  - Submits the dataset and its corresponding metadata to the **data repository**.
  - Generates a descriptive **front page** summarizing the dataset's details and FAIR status.
- If the dataset is found to be partially non-compliant:
  - The system generates a set of **FAIR improvement guidelines** to assist the user.
  - Both the guidelines and dataset are stored within the **repository** for reference and traceability.
- The system architecture comprises several core components:
  - **Client User**—the researcher or contributor submitting the dataset.
  - **FAIR Service**—responsible for dataset evaluation and form generation.
  - **Server Hosting the Web Service**—handles communication between the platform and the repository.
  - **Data Repository** – stores datasets, metadata, and generated FAIR guidelines.
- Communication between all modules is implemented using **RESTful web services and APIs** to ensure flexibility and interoperability.

- Upon completion, the service provides the user with a detailed **FAIR compliance report** and recommendations for further improvement.

Overall, this workflow demonstrates a semi-automated approach to improving data FAIRness within the SoBigData ecosystem. By combining a decision-tree-based questionnaire with automated evaluation and feedback, the FAIR Service effectively supports researchers in maintaining transparent, reusable, and well-documented datasets.

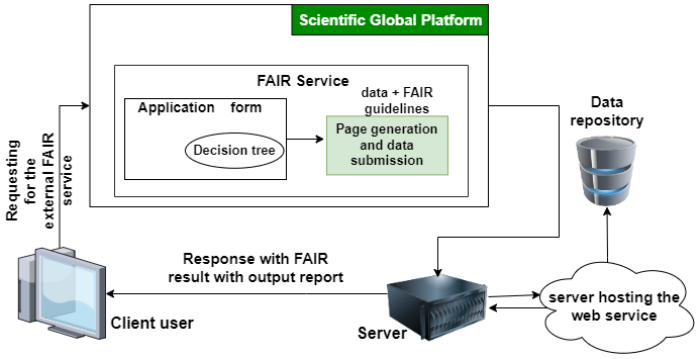


FIGURE 7.5: Proposed workflow of the FAIR Service for improving FAIR principles within the SoBigData repository.

7.4.2 Approaching the idea with our model FAIR-IS-ON

Following the approach shown in Figure 7.5, a prototype model named **FAIR-IS-ON** was designed to semi-automate the FAIR assessment process. Although the full implementation is still under development, the model was locally prepared and tested on a Windows-based environment to demonstrate its feasibility.

The basic architecture of FAIR-IS-ON, shown in Figure 7.6, outlines the key components and workflow:

- The system is designed to operate through multiple **client interfaces**, including web-based applications, desktop platforms, and mobile devices (Android and iPhone).
- Clients send **HTTP requests** to the server, which processes them through the web service layer.
- The **web server** consists of:
  - **Apache** for managing HTTP communication and request handling.
  - A **REST API** responsible for data exchange between client and database.
  - Additional modules for **authentication** and a **load balancer** to ensure secure and efficient communication.
- The backend uses a **MySQL database** to store user inputs, FAIR assessment results, and generated recommendations.
- Once processed, the server sends a **plain-text response** (such as assessment reports or feedback) back to the client interface.
- This architecture supports cross-platform integration and can be scaled for a full web-based FAIR evaluation service in future iterations.

In its current stage, the FAIR-IS-ON system operates in a semi-automated manner on a local machine, demonstrating the foundational workflow for future implementation of a fully web-integrated FAIR evaluation platform.

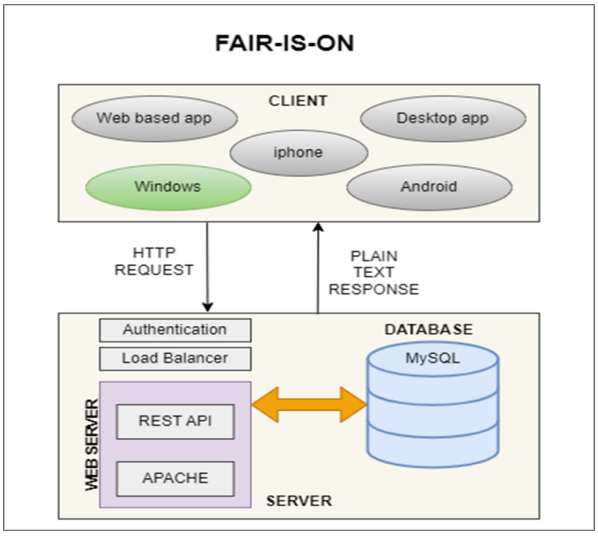


FIGURE 7.6: The proposed FAIR-IS-ON tool represents a semi-automated web-based service that is presently in the development phase.

7.4.3 System Architecture and Technical Implementation

Overview

The FAIR-IS-ON system has been implemented as a web-based application designed to evaluate and display FAIR (Findability, Accessibility, Interoperability, and Reusability) assessment results. The architecture follows a client-side rendering model using **ReactJS** and exchanges data in **JSON** format. The system is lightweight, modular, and scalable, making it suitable for FAIR data evaluation and visualization workflows.

Backend Architecture

The backend of the FAIR-IS-ON model is implemented using **Java (version 17)** and the **Spring Boot (version 3.1.5)** framework. It is designed to handle the data storage, survey management, and report generation components of the system. The backend interacts with a relational database to manage information associated with the four FAIR principles and user-generated reports.

- **Platform:** IntelliJ IDEA
- **Architecture pattern:** Layered Spring Boot architecture (Controller → Service → Repository → Database)
- **Controllers:** Two controllers—one for managing survey questions and another for handling report operations.

- **Services and Repositories:** Five services and repositories (four for FAIR principle tables and one for report management) implementing CRUD[5] (Create, Read, Update, Delete) operations using Spring Data JPA.
  - **Create** Adds new data to the system *SQL:* INSERT *HTTP:* POST
  - **Read** Retrieves existing data without modification *SQL:* SELECT *HTTP:* GET
  - **Update** Modifies existing data *SQL:* UPDATE *HTTP:* PUT / PATCH
  - **Delete** Removes data from the system *SQL:* DELETE *HTTP:* DELETE
- **Exception Handling:** A custom `RecordNotFoundException` is implemented for missing records.

The backend manages five main tables:

1. `survey_questions`, `survey_questions1`, `survey_questions2`, `survey_questions3` – storing questions for each FAIR principle.
2. `survey_report` – storing generated reports.

Each survey question table uses a composite primary key (`surveyId, quesId`), whereas the report table uses an auto-incrementing primary key (`id`). All data is exchanged between the frontend and backend as **JSON payloads** through REST APIs.

LISTING 7.1: Example REST endpoint for survey question retrieval

```
@RestController
@RequestMapping("/api/surveys")
public class SurveyQuestionController {
  @Autowired
  private SurveyQuestionsService service;

  @GetMapping("/{surveyId}/questions/{quesId}")
  public SurveyQuestions getQuestion(
        @PathVariable int surveyId, @PathVariable int quesId) {
    return service.getQuestion(surveyId, quesId);
  }
}
```

Frontend Framework

The frontend is developed using **React (version 18)**, a JavaScript library for building reusable user interface components. It manages the state of the multi-step evaluation form and dynamically updates the interface without page reloads. The front page has been shown in Figure 7.7

- **Libraries used:** `react`, `react-dom`
- **Framework setup:** Initialized with `Create React App` (`react-scripts`), providing Webpack and Babel integration for code bundling and transpilation.
- **Routing:** Implemented through `react-router-dom (v6)` to manage navigation between steps of the FAIR evaluation process, such as Input, Validation, Evaluation, and Review.
- **Structure:** Component-based modular development ensuring separation between UI, logic, and data management.

[5] `https://www.crowdstrike.com/en-us/cybersecurity-101/observability/crud/`

Data Handling and JSON Integration

All application data is managed using **JavaScript Object Notation (JSON)**, ensuring lightweight and interoperable communication between components. Form data collected at each step is stored in structured JSON objects, shared via the React Context API and `useState` hooks. At the final stage, the data is serialized into JSON for export or further processing.

LISTING 7.2: Example JSON structure used in FAIR-IS-ON

```
{
  "dataset_id": "doi:10.5281/zenodo.12345",
  "tool_used": "FAIR-Aware",
  "metrics": {
    "Findability": 7.5,
    "Accessibility": 6.0,
    "Interoperability": 5.0,
    "Reusability": 8.0
  },
  "overall_score": 6.6
}
```

Styling and Design

The application uses **Cascading Style Sheets (CSS)** for layout, spacing, and responsive design. Global styles are defined in `index.css`, and modular CSS files are used for component-specific styling.

- **Styling approach:** Global CSS combined with modular component styles.
- **Design characteristics:** Clean, minimal layout emphasizing accessibility and clarity.
- **Responsive behavior:** Media queries ensure appropriate rendering across devices.

LISTING 7.3: Sample CSS styling snippet

```
body {
  font-family: Arial, sans-serif;
  background: #fafafa;
}
form {
  max-width: 500px;
  margin: auto;
  background: #fff;
  padding: 24px;
  border-radius: 12px;
}
```

Application Workflow

The FAIR-IS-ON model follows a multi-step user interaction workflow implemented using `react-router-dom`:

- **Step 1 – Input Collection:** Users provide dataset identifiers and metadata.
- **Step 2 – Evaluation Configuration:** Users select FAIR assessment parameters and tools.

- **Step 3 – Review and Submission:** The system compiles entries into a single JSON file for export or visualization.
- **Step 4 – Visualization (optional):** The results can be viewed as structured summaries or JSON previews.

This modular workflow improves navigation and allows users to move between steps without losing data. Figure 7.7 displays the tool's front page for the client.

Testing and Performance Evaluation

Testing was conducted using the **React Testing Library** and **Jest-DOM** to simulate user interactions and validate UI components. Performance metrics such as loading time and visual stability were monitored using the **Web Vitals** library. The production build was optimized with `react-scripts build` for faster load times.

Deployment and Environment

The application runs in a **Node.js** environment during development and can be deployed on any modern web server such as *Vercel* or *Netlify*. Build tools such as Webpack and Babel (provided by `react-scripts`) manage bundling and transpilation. Browser compatibility is ensured using the Browserslist configuration included in the project.

Summary

In summary, the FAIR-IS-ON architecture employs a **React-based frontend**, **JSON-based data management**, and **CSS-driven design** to provide an interactive and responsive environment for the FAIR-a tool evaluation. This structure ensures modularity, maintainability, and compliance with modern web development standards while remaining lightweight and extensible for future integration with FAIR APIs or backend evaluation engines.

FIGURE 7.7: The front welcome page of our proposed semi-automated web-based FAIR-IS-ON model

Chapter 8

Evaluation of FAIR assessment Tools

The evaluation procedure for FAIR assessment tools, or FAIR-a tools, is described in this chapter. The chapter is divided into some sections to understand clearly. Section 8.1 highlights the overall evaluation procedure, while Section 8.2 highlights the assessment methodology. In contrast, the selection criteria and background for the FAIR-a tools utilized in this assessment are explained in Section 8.3. Section 8.4 notes the experimental settings for the FAIR-a tools assessment process, whereas Section 8.5 focuses on the evaluation process's outcomes and the relevant tools that need to have been employed. Lastly, Section 8.6 highlights the outcomes and limits of this assessment, and Section 8.7 provides the chapter's conclusions. The assessment process highlights the usability, degree of automation for FAIR-a tools, and degree to which the tools provide standards for interoperability.

8.1 Evaluation Process

Developing a clear direction for the evaluation process required formulating a set of research questions that address multiple dimensions of the FAIR principles and their role within open science platforms. These questions ensure a systematic approach to understanding how FAIR assessment tools enhance the FAIRness, accessibility, and overall usability of research data.

Specifically, four research questions (RQs) were formulated to explore the relationship between FAIR principles and open science practices, define evaluation criteria, assess the performance of existing FAIR-a tools, and identify their key limitations and potential improvements. Together, these questions provide a structured foundation for the analysis and support the development of a comprehensive FAIR assessment framework.

- **RQ1: What evaluation dimensions and criteria are required to design a systematic framework for assessing FAIR assessment tools within Open Science Platforms?**
  This question focuses on identifying the key evaluation dimensions and criteria required to design a systematic framework for assessing FAIR assessment tools. It aims to establish a structured foundation that enables consistent and comprehensive evaluation across different tools within open science platforms.

- **RQ2: How can the proposed evaluation framework be applied to assess the implementation of FAIR principles within open science platforms?**

This question examines how the proposed evaluation framework can be applied to assess the implementation of FAIR principles within open science platforms. It emphasizes the practical use of the framework to enable structured comparison and analysis across different FAIR assessment tools.

- **RQ3: How effective are existing FAIR assessment tools in measuring the FAIRness maturity of OSPs and identifying areas for improvement?**
  This question evaluates existing FAIR assessment tools in terms of their effectiveness in measuring FAIRness maturity within open science platforms. It analyzes how well these tools assess usability, runtime performance, and interpretability, while identifying strengths and weaknesses in datasets, repositories, and digital objects.
- **RQ4: What are the main limitations of FAIR-a tools, and what strategies can be proposed to improve their usability and reliability?**
  This question aims to identify the main challenges and limitations associated with existing FAIR assessment tools. It further explores potential strategies to improve their usability, interpretability, and reliability, enhancing their effectiveness for researchers and data managers.

These research questions collectively guide the evaluation process by linking the theoretical foundations of FAIR principles with their practical application in open science platforms. They serve as the basis for the methodology and analysis presented in the subsequent sections.

Hardware and Software Settings

The evaluation of the selected FAIR-a tools was performed in a consistent environment to ensure fairness and reproducibility. The following system configuration and settings were used during the assessment and are shown in Table 8.1 below.

Parameter	Description
Operating System	Windows 11 (64-bit) standard desktop environment.
Processor	Intel Core i5, 2.4 GHz quad-core CPU.
Memory (RAM)	8 GB.
Internet Connection	Stable broadband connection used for accessing online tools.
Browser	Google Chrome (latest version) for all web-based FAIR-a tools.
Execution Type	Manual operation; each tool accessed through its web interface or downloadable version.
Dataset Input	Example datasets and metadata files are manually uploaded or linked for testing FAIR compliance.
Evaluation Metrics	Time taken for execution, adaptability, effort cost, and usability level were recorded for each tool.
Observation Method	Results and user experience were documented immediately after each execution.

TABLE 8.1: Experimental setup and evaluation conditions

All evaluations were conducted on the same system to maintain consistency in timing, performance observations, and usability comparisons across different FAIR-a tools. In this study, the 'user's assessment' perspective is based on the evaluation

experience of our testing team; therefore, the term "user" is used to represent this viewpoint.

8.2 Methodology

We proposed a comprehensive, bottom-up evaluation criteria aimed at investigating inconsistencies across FAIR-a tools following the Krans et al.'s work [158], which evaluated ten FAIR assessment (FAIR-a) tools for chemical and nano-material safety

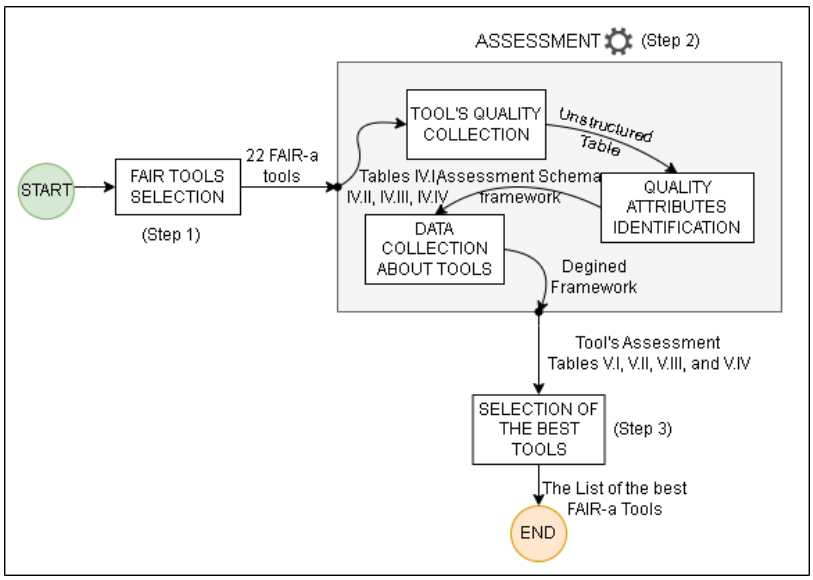


FIGURE 8.1: FAIR-a tools evaluation methodology

Figure 8.1 illustrates the methodology adopted in this study. In the first step, *FAIR TOOLS SELECTION*, we identified 22 publicly available FAIR assessment (FAIR-a) tools. This was followed by the *TOOL'S QUALITY COLLECTION* step, where we gathered qualitative and quantitative data about the tools and organized it in an initial unstructured table. Next, in the *QUALITY ATTRIBUTES IDENTIFICATION* phase, we analyzed this data to derive key evaluation dimensions, which informed the development of our assessment schema, as presented in Tables 8.3, 8.4, 8.5, and 8.6.

The *DATA COLLECTION ABOUT TOOLS* step involved populating the evaluation framework based on the identified quality attributes. The results of this evaluation are presented in Tables 8.7, 8.8, 8.9, and 8.10. Finally, the *SELECTION OF THE BEST TOOLS* step synthesizes the findings to produce a ranked list of the most effective FAIR-a tools.

For the assessment process, we used the SoBigData[1] research infrastructure as our Open Science Platform (OSP). SoBigData RI is particularly well-suited for evaluating FAIR compliance, as it is designed to support open science through the publication of data and artifacts that adhere to FAIR principles. Additionally, we utilized the dataset by Hasnain and Rebholz-Schuhmann [166][2] to conduct FAIRness assessments.

[1] `https://sobigdata.d4science.org/`

[2] `https://physionet.org/content/mmash/1.0.0/` — Version 1.0.0 released on June 19, 2020

8.3 FAIR-a Tools selection

This section details the process we followed to find FAIR-a tools.

8.3.1 FAIR-a Tool's findings

Among the FAIR-a tools available in the *fairassist dataset*[3], we selected 22 tools that were accessible and executable at the time of evaluation. And the FAIRassist Registry[4] is a curated registry of FAIR assessment metrics, benchmarks, and tools for the four FAIR foundational principles—Findable, Accessible, Interoperable, and Reusable. An interactive interface is employed in navigating hierarchical representations of sub-principles (F1–F4, A1–A2, I1–I3, R1–R1.3) and viewing how various assessment tools instantiate these indicators. The site enables filtering by object type, subject domain, organization, or specific tool and shows an unobstructed view of what metrics each tool deploys. Such a registry is critical for translating conceptual FAIR principles into quantifiable criteria and enabling comparability between assessment approaches. In this study, the FAIRassist registry was employed to identify and validate candidate FAIR-a tools. Its structured representation of metrics ensured that the 22 FAIR-a tools selected for analysis encompassed a diverse range of FAIR indicators, enabling a comprehensive evaluation across functional, technical, runtime, and usability aspects.

These tools are described in terms of *resources, execution types, key features, organizations, target objects, and reading materials*. For each, FAIR-a Tool's links are provided for further exploration, as shown in Table 8.2, along with relevant publications listed under reading materials, making this an efficient way to access comprehensive information on FAIR-a Tools in one place.

A systematic summary of the main FAIR assessment (FAIR-a) tools is provided in Table 8.2, arranged according to the type of interaction and level of automation. Online Survey-Based Self-Assessment Tools (OSBSAT) are tools that lead users through structured questionnaires; Online Semi-Automated Tools (OSAT) combine automated scoring mechanisms with user inputs; and Online Automated Tools (OAT) use machine-readable metadata to assess FAIRness. Furthermore, the table includes Other Types (OT) that support more general FAIR-related tasks like data stewardship planning and training, as well as Offline Self-Assessment Survey-Based Tools (OFSBSAT), which usually involve manual evaluation using downloadable templates. The table includes the tool name, the accountable organization, and a direct access link for every tool. This category demonstrates the variety of current FAIR assessment methods created by international projects, research infrastructures, and open science communities.

It is worth mentioning that many of these tools are still under development, so their long-term stability and availability remain uncertain.

8.3.2 FAIR-a Tool's Description

We have created a GitHub repository[5] that includes additional materials pertaining to the evaluation of FAIR-a tools in order to promote transparency and reproducibility. This includes a design overview of our evaluation framework (`FRAMEWORK design.pdf`), structured summary tables (`FAIR tables for upload.pdf`, `Tools Name`

[3] `https://fairassist.org/`

[4] `https://fairassist.org/registry`

[5] `https://github.com/PayelPatra/FAIR_project.git`

`.pdf`), the qualitative assessment results (`The Quality Performance of FAIR tools .pdf`), and detailed tool descriptions (`FAIR Tool explanation.txt`). The repository also includes a documented discussion from the first review process (`Result & Discussion from team's review.txt`) and a `README.md` file for navigation. The repository functions as an extended reference to facilitate additional exploration and reuse, even though Table 8.2 provides a summary of the main tool categories and access links. To keep the paper self-contained, every requirement has been included.

8.4 Assessment Settings of FAIR-a Tools'

This section provides the background of our study, organized into the following subsections: *FAIR-a Tool's Quality Collection, FAIR-a Tool's Quality Attribute Identification, and FAIR-a Tool's' Data Collection*. Section 8.4.1 outlines the preparatory setup of FAIR-a Tool's quality collection, which is required before beginning tool evaluations. Section 8.4.2 discusses the assessment of tool performance, while Section 8.4.3 introduces the data collection of FAIR-a tools, including the research questions proposed for essential criteria and the FAIR Principle's support of FAIR-a tools in OSP.

8.4.1 Tool's Quality Collection

To conduct the evaluation of FAIR-a tools, we introduced five key groups, listed in Table 8.2, in which FAIR-a tools are categorized: *(i)* Online survey-based self-assessment types **(OSBSAT)**, which include web-based tools designed with standard web technologies; *(ii)* Semi-automated Types **(SAT)**, which combine online accessibility with more interactive evaluation logic; *(iii)* Online automated Types **(OAT)**, which run automated FAIRness checks, some requiring local setup while others are accessible via direct web interfaces; *(iv)* Offline survey-based self-assessment types **(OFSBSAT)**, which are downloadable tools used offline; and *(v)* Other types **(OT)**, which do not fit into the above categories but still provide FAIR-related assessments.

Each category has distinct evaluation characteristics, ranging from simple browser-based accessibility to advanced automated testing.

Dataset & FAIR-a Tool's Selection

Data Findings We selected the *SoBigData*[6] repository for this study,y as it is part of the European research infrastructure explicitly designed to support data science under the FAIR and FACT principles, ensuring ethical, privacy-preserving, and reproducible practices [18], [25]. Within this repository, we focused on the health domain, where rigorous FAIR practices are especially crucial.

Specifically, we chose the **Multilevel Monitoring of Activity and Sleep in Healthy People (MMASH)** dataset[7], which provides 24-hour continuous data for 22 healthy participants. The structured and comprehensive nature of MMASH makes it well aligned with our FAIR assessment goals, serving as an ideal case study for testing and validating FAIR-a tools.

Selection of FAIR-a Tools We worked with **22 FAIR-a tools**, each possessing distinct characteristics and distributed across the five categories shown in Table 8.2.

[6] `www.sobigdata.eu`

[7] `https://physionet.org/content/mmash/1.0.0/`

Seven tools fall under the **OSBSAT** group *[FAIR Data Self-Assessment Tool, FAIR-Aware, FAIRdat, FairDataBR, NFDI4Culture FAIR-Check, SATIFYD]*, which are browser-based and straightforward to test. The **SAT** group includes *FAIRshake*, which evaluates the FAIRness of research digital resources via a web interface without requiring local setup.

The **OAT** group consists of nine tools *[F-UJI, FAIR EVA, FAIR Evaluator, FAIR enough, FAIR-Checker, FOOPS!, HowFAIRIs, O'FAIRe, OpenAIRE Validator - FAIR assessment]*. While most are accessible online, some, such as *FAIR EVA* and *O'FAIRe* require a local installation (e.g., via Docker or Java-based deployment). The remaining seven are directly usable online and require only an internet connection. These automated tests usually take about 15 minutes or more to complete. However, we faced technical difficulties with *FAIR Evaluator* and *OpenAIRE Validator - FAIR assessment*, limiting the results obtained from them.

The remaining tools are distributed across the **OFSBSAT** and **OT** groups, completing the total of 22 tools evaluated in our framework.

In summary, while our evaluation focused on 22 widely used FAIR-a tools, we acknowledge that additional tools exist beyond this list, some of which are under development or face technical limitations. This careful selection ensures that our framework captures both the diversity and the evolving nature of FAIR tools in scientific research.

8.4.2 FAIR-a Tools' Quality Attribute Identification

Tool's Quality Setup

In our combined evaluation framework *(Functionality-related & Technical-related)* and performance framework *(Runtime Aspects & Usability and User Satisfaction)* (Tables 8.3, 8.4, 8.5, 8.6), we evaluated several quality aspects of the tools:

(i) **Adaptability (Extensibility)** (ii) **Degree of Maturity** (iii) **Qualitative/Quantitative Result** (iv) **Development State** (v) **Focused Principle** (vi) **FAIR Strategy (Assessment/Improvement)** (vii) **Effort Cost of Using Tool** (viii) **Evaluation Grade (Good/Better/Best)** (ix) **FAIR Improvement Scope**

Detailed descriptions of this framework are explained in Section 8.4.3, and the complete evaluation results are available in our GitHub repository[8].

8.4.3 FAIR-a Tools' Data Collection

This subsection introduces how FAIR-a tools were assessed from a data collection perspective. We focus first on their performance characteristics, then define evaluation guidelines, and finally address the research questions (RQ1 and RQ2) through a classification and detailed analysis of selected tools.

Performance Checking

In order to understand the practical reliability of FAIR-a tools, we evaluated their performance under typical usage conditions. The analysis concentrated on three core aspects, which help determine whether a tool is efficient, consistent, and robust.

Response time: the speed at which the FAIR-a tool reacts to a user request or input. Quicker reaction times enhance productivity and user satisfaction.

[8] `https://github.com/PayelPatra/FAIR_project.git`

TABLE 8.2: Overview of the FAIR Assessment (FAIR-a) Tool's information, including website links for each Key Group category: Online Survey-Based Self-Assessment Tool (OSBSAT), Online Semi-Automated Tool (OSAT), Online Automated Tool (OAT), Offline Self-Assessment Survey-Based Tool (OFSBSAT), and Other Types (OT).

Key Group	Tool Name	Organization	Website Link
OSBSAT	FAIR Data Self-Assessment Tool	ARDC	https://ardc.edu.au/resource/fair-data-self-assessment-tool/
OSBSAT	FAIR Data Self-Assessment Tool	Cross-Lateral Enterprises Pty Ltd	https://crosslateral.com.au/fair-tool.html
OSBSAT	FAIR-Aware	FAIRsFAIR (DANS)	https://fairaware.dans.knaw.nl/
OSBSAT	FAIRdat	DANS	https://www.surveymonkey.com/r/fairdat
OSBSAT	FairDataBR	UFPB	https://wrco.ufpb.br/fair/index.html
OSBSAT	NFDI4Culture FAIR-Check	NFDI4Culture	https://nfdi4culture.de/services/fair-check
OSBSAT	SATIFYD	DANS	https://satifyd.dans.knaw.nl/
OSAT	FAIRshake	NIH Data Commons Partners	https://fairshake.cloud/
OAT	F-UJI	FAIRsFAIR (PANGAEA)	https://github.com/pangaea-data-publisher/fuji
OAT	FAIR EVA	CSIC	https://github.com/EOSC-synergy/FAIR_eva
OAT	FAIR Evaluator	FAIRmetrics.org	https://fairsharing.github.io/FAIR-Evaluator-FrontEnd/#!/
OAT	FAIR Enough	Maastricht University	https://fair-enough.semanticscience.org/
OAT	FAIR-Checker	IFB (ELIXIR-FR)	https://fair-checker.france-bioinformatique.fr/
OAT	FOOPS!	Universidad Politécnica de Madrid	https://foops.linkeddata.es/FAIR_validator.html
OAT	HowFAIRIs	Netherlands eScience Center	https://pypi.org/project/howfairis/
OAT	O'FAIRe	AgroPortal Project (INRAE / U. Montpellier)	https://github.com/agroportal/fairness
OAT	OpenAIRE Validator – FAIR assessment	OpenAIRE	https://provide.openaire.eu/home
OFSBSAT	Do I-PASS for FAIR	LCRDM Task Group	https://zenodo.org/records/4080867
OFSBSAT	FAIRness Self-Assessment Grids	RDA-SHARC IG	https://zenodo.org/records/3922069
OT	Data Stewardship Wizard	ELIXIR CZ and ELIXIR NL	https://ds-wizard.org/
OT	TRIPLE Training Toolkit	TRIPLE Project (CNR lead)	https://project.gotriple.eu/project-deliverables/triple-training-toolkit/
OT	FAIRplus	Innovative Medicines Initiative (IMI)	https://fairplus-project.eu/

Reliability: the consistency with which the FAIR-a tool performs without errors, interruptions, or disruptions, thereby guaranteeing trustworthy results.

Failure in one-time testing: whether the tool malfunctions during a single run, including issues such as crashes or incorrect outputs.

This performance evaluation revealed both strengths and limitations across the tested tools, forming the basis for integrating quality aspects into our broader framework (Tables 8.3, 8.4, 8.5, 8.6). Detailed documentation is provided in our GitHub repository.

Evaluation Guidelines

To ensure comparability and rigor, we established a set of evaluation guidelines. These guidelines formalize the assessment process and describe the framework attributes used across all tool groups.

As noted in [158], some criteria align with prior studies on tool evaluation frameworks. In our case, evaluation included adherence to FAIR principles, documentation quality, usability, interpretability, and both qualitative and quantitative observations.

Grouping Criteria We classified the 22 tools into five categories (OSBSAT, OSAT, OAT, OFSBSAT, and OT, defined in Section 8.4.1). Grouping was based on features such as **Input Type, Data Type, FAIR Strategy, Target Object, and Applicability**. This categorization highlights technological differences, guidance levels, usability, effort requirements, and assessment durations.

Description of the Framework Parameters The framework includes attributes related to functionality, technical robustness, runtime aspects, and usability.

TABLE 8.3: Proposed Evaluation Framework for FAIR-a Tools Based on Functionality. Note: The listed input types, target objects, principles, and result types represent the range observed across all tool groups. They are not exclusive to a single tool, and some tools may support only a subset of these items.

Functionality-Related	**Tool Groups**	OSBSAT, OSAT, OAT, OFSBSAT, OT
	Input types	Multiple choice, radio button, input box, URL/PID, DOI/GUID, digital objects, resource, software, repository, web questionnaire, manual checklist
	Target objects	Dataset, all digital objects, terminology artifacts, people's knowledge, software
	Focused principle	Findable, Accessible, Interoperable, Reusable
	FAIR improvement scope	Yes with descriptive text, No
	Result type	Qualitative, Quantitative

TABLE 8.4: Proposed Evaluation Framework for FAIR-a Tools Based on Technical Aspects.

Technical Related	**Tool Groups**	OSBSAT, OSAT, OAT, OFSBSAT, OT
	Degree of Maturity	Yes, No
	Problems to Fix	Yes, Yes with explanation, No
	Development State	Developed, Under Developed

TABLE 8.5: Proposed Evaluation Framework for FAIR-a Tools Based on Runtime Aspects.

Runtime Aspects	**Tool Groups**	OSBSAT, OSAT, OAT, OFSBSAT, OT
	Adaptability (Extensibility)	Yes, No
	Effort Cost of Using Tools	Low, Medium, High
	Execution Time	Minutes

TABLE 8.6: Proposed Evaluation Framework for FAIR-a Tools Based on Usability and User Satisfaction.

Usability and User Satisfaction	**Tool Groups**	OSBSAT, OSAT, OAT, OFSBSAT, OT
	Level of Understanding Question	Easy, Medium, Hard
	Level of Understanding Output	Easy, Medium, Hard
	Level of Tool's Usability	Easy, Medium, Hard
	User Experience	Best, Better, Good
	Recommend to Others	Yes, Partial, No
	Overall Evaluation Grade	Good, Better, Best

Together, Tables 8.3–8.6 present a comprehensive evaluation framework for FAIR-a tools, structured across four complementary dimensions. The functionality-related aspects describe the types of inputs, target objects, FAIR principles addressed, and the nature of assessment results. The technical-related aspects capture the maturity level, development status, and identified issues associated with the tools. Runtime-related aspects focus on adaptability, effort required for tool usage, and execution time, reflecting practical deployment considerations. Finally, usability and user-satisfaction aspects evaluate user understanding, tool usability, user experience, and overall perception, providing insight into the effectiveness of the tools from an end-user perspective.

Addressing RQ1: Evaluation Framework Specification

To answer RQ1, we have taken care of how the FAIR principles support the advancement of open science. By ensuring that research artifacts are **Findable**, **Accessible**, **Interoperable**, and **Reusable**, these principles foster transparency, reproducibility, and collaboration across scientific communities. They also strengthen trust and accountability by enabling reliable data sharing and reuse, which are cornerstones of open science.

Building on this foundation, we specified an evaluation framework that organizes FAIR-a tool's characteristics—into five categories. This framework ensures that both technical and usability-related aspects are captured in a structured way.

TABLE 8.7: Proposed Evaluation Framework based on Functionality.

Functionality-Related					
Tool / Group	**IT**	**TO**	**FP**	**FIS**	**RT**
Key Group 1 — Online Survey-Based Self-Assessment					
FAIR Data Self-Assessment Tool	Multiple choice	Datasets	F, A, I	Understanding about FAIR.	Qualitative
FAIR Data Self-Assessment Tool	Multiple choice	Datasets	F, A, I	Knowledge enhancement.	Quantitative
Fair-Aware	Radio	People's knowledge	F, A, I, R	Knowledge about FAIR principles.	Quantitative
FAIRdat	Radio, input textbox	Datasets	F, A, I	Understanding about FAIR.	Qualitative
FairDataBR	Radio, multiple choice	Datasets	F, A, I, R	Examine if data is FAIR.	Quantitative
NFDI4Culture FAIR-Check	Multiple choice	Datasets	F, A, I, R	Understand about FAIR.	Qualitative
SATIFYD	Multiple choice	Datasets	F, A, I, R	Knowledge and check FAIR data.	Quantitative
Key Group 2 — Online Semi-Automated					
FAIRshake	Multiple choice	All digital objects	F, A, I, R	No	Colored rubric (quantitative)
Key Group 3 — Online-Automated					
F-UJI	PID/URL	Datasets	F, A, I, R	Yes	Quantitative
FAIR EVA	DOI/PID	Datasets	F, A, I, R	Yes	Quantitative
FAIR Evaluator	URL, maturity indicator, GUID	All digital objects	F, A, I, R	Yes	Quantitative
FAIR Enough	URL, digital object	All digital objects	F, A, I, R	Yes	Quantitative
FAIR-Checker	URL/DOI	All digital objects	F, A, I, R	Yes	Quantitative
FOOPS!	URI	Terminology artifacts	F, A, I, R	Yes	Quantitative
HowFAIRIs	Software installation	Software	F, A, I, R	Yes	Quantitative
O'FAIRe	Access rights	Terminology artifacts	F, A, I, R	Yes	Quantitative
OpenAIRE Validator	Repository	All digital objects	F, A, I, R	Yes	Quantitative
Key Group 4 — Offline Survey-Based Self-Assessment					
Do I-PASS for FAIR	Multiple choice, comments	Organizations	F, A, I, R	No	Qualitative
FAIRness Self-Assessment Grids	Input "1"	All digital objects	F, A, I, R	No	Quantitative
Key Group 5 — Other Types					
Data Stewardship Wizard	Web / manual questionnaires	All digital objects	F, A, I, R	No	NA
TRIPLE Training Toolkit	Manual checklist	All digital objects	F, A, I, R	No	NA
FAIRplus	Manual checklist	All digital objects	F, A, I, R	No	Quantitative

Note: IT = Input Types; TO = Target Objects; FP = Focused Principle; FIS = FAIR Improvement Scope; RT = Result Type.

TABLE 8.8: Proposed Evaluation Framework based on Technical Aspects.

	Technical-Related		
Tool Group	**DM**	**PF**	**DS**
Key Group 1 — Online Survey-Based Self-Assessment			
FAIR Data Self-Assessment Tool	Yes	Yes	Developed
FAIR data Self-Assessment Tool	Yes	No	Developed
Fair-Aware	Yes	No	Developed
FAIRdat	No	Yes	Developed
FairDataBR	No	Yes	Developed
NFDI4Culture FAIR-Check	No	No	Developed
SATIFYD	Yes	No	Developed
Key Group 2 — Online Semi-Automated			
FAIRshake	No	No	Developed
Key Group 3 — Online-Automated			
F-UJI	Yes	No	Under Developed
FAIR EVA	No	Yes	Developed
FAIR Evaluator	No	No	Developed
FAIR enough	Yes	No	Developed
FAIR-Checker	Yes	No	Under Developed
FOOPS!	No	Yes	Under Developed
HowFAIRIs	Yes	Yes	Under Developed
O'FAIRe	No	No	Developed
OpenAIRE Validator	Yes	Yes	Developed
Key Group 4 — Offline Survey-Based Self-Assessment			
Do I-PASS for FAIR	No	No	Developed
FAIRness self-assessment grids	No	No	Developed
Key Group 5 — Other Types			
Data Stewardship Wizard	Yes	No	Developed
TRIPLE Training Toolkit	No	No	Developed
FAIRplus	No	No	Developed

Note: DM = Degree of Maturity; PF = Problems to Fix; DS = Development State.

TABLE 8.9: Proposed Evaluation Framework based on Runtime aspects.

	Runtime-Aspects		
Tool Group	**AD (EX)**	**EC**	**ET**
Key Group 1 — Online Survey-Based Self-Assessment			
FAIR Data Self-Assessment Tool	No	Low	10
FAIR data Self-Assessment Tool	No	Low	11.36
Fair-Aware	No	Low	14.73
FAIRdat	No	Low	9.36
FairDataBR	No	Low	12.56
NFDI4Culture FAIR-Check	No	Medium	14.28
SATIFYD	No	Low	11
Key Group 2 — Online Semi-Automated			
FAIRshake	Yes	High	20.73
Key Group 3 — Online-Automated			
F-UJI	Yes	Low	9
FAIR EVA	Yes	Low	76.9
FAIR Evaluator	No	Medium	48.48
FAIR enough	No	Low	1.78
FAIR-Checker	No	Low	27.7
FOOPS!	No	Low	0.92
HowFAIRIs	Yes	Medium	10.16
O'FAIRe	No	Medium	14
OpenAIRE Validator	No	Medium	10.68
Key Group 4 — Offline Survey-Based Self-Assessment			
Do I-PASS for FAIR	No	Medium	22.05
FAIRness self-assessment grids	No	Medium	49.76
Key Group 5 — Other Types			
Data Stewardship Wizard	No	Medium	49
TRIPLE Training Toolkit	No	Medium	70
FAIRplus	No	Medium	15.36

Note: AD = Adaptability; EX = Extensibility; EC = Effort Cost of Using Tools; ET = Execution Time (minutes).

TABLE 8.10: Proposed Evaluation Framework based on Usability and User Satisfaction.

Usability and User Satisfaction-Related						
Tool / Group	**UQ**	**UO**	**TU**	**UX**	**RO**	**OG**
Key Group 1 — Online Survey-Based Self-Assessment						
FAIR Data Self-Assessment Tool	Medium	Easy	Easy	Good	Partial	Better
FAIR data Self-Assessment Tool	Medium	Easy	Easy	Best	Yes	Best
Fair-Aware	Easy	Easy	Easy	Best	Yes	Best
FAIRdat	Easy	Easy	Easy	Good	Partial	Better
FairDataBR	Hard	Medium	Easy	Good	Partial	Better
NFDI4Culture FAIR-Check	Easy	Easy	Easy	Good	Partial	Better
SATIFYD	Easy	Easy	Easy	Best	Yes	Best
Key Group 2 — Online Semi-Automated						
FAIRshake	Hard	Tough	Hard	Good	No	Good
Key Group 3 — Online-Automated						
F-UJI	Easy	Easy	Easy	Best	Yes	Best
FAIR EVA	Medium	Easy	Hard	Good	No	Good
FAIR Evaluator	Medium	Medium	Hard	Good	No	Good
FAIR enough	Easy	Easy	Easy	Best	Yes	Best
FAIR-Checker	Easy	Easy	Easy	Best	Yes	Best
FOOPS!	Easy	Easy	Easy	Better	Partial	Better
HowFAIRIs	Medium	Medium	Medium	Better	Partial	Better
O'FAIRe	Medium	Medium	Medium	Good	No	Good
OpenAIRE Validator	Medium	Medium	Medium	Good	No	Good
Key Group 4 — Offline Survey-Based Self-Assessment						
Do I-PASS for FAIR	Medium	Easy	Easy	Better	Partial	Better
FAIRness self-assessment grids	Hard	Medium	Easy	Good	No	Good
Key Group 5 — Other Types						
Data Stewardship Wizard	Medium	Easy	Tough	Good	No	Good
TRIPLE Training Toolkit	Medium	Easy	Medium	Good	No	Good
FAIRplus	Medium	Easy	Easy	Better	No	Good

Note: UQ = Understanding Questions; UO = Understanding Output; TU = Tool Usability; UX = User Experience; RO = Recommend to Others; OG = Overall Grade.

By applying this framework, we can identify critical attributes in existing tools, guide the selection of suitable FAIR-a tools, highlight research gaps, and inspire the design of new, user-friendly tools incorporating advanced techniques.

Answer to RQ1: A multidimensional evaluation framework is proposed for FAIR assessment tools, structured around key dimensions including functionality, technical robustness, runtime behavior, and usability. Within these dimensions, specific evaluation criteria are defined to enable systematic, consistent, and reproducible assessment. This framework provides a standardized basis for comparing FAIR tools and assessing FAIRness within open science platforms.

Addressing RQ2: Using the Evaluation Framework to Classify Existing FAIR-a Tools

To answer RQ2, we defined the essential criteria for evaluating the implementation of FAIR principles within OSPs using the framework introduced in RQ1. These criteria include **Input Type, Target Object, FAIR Strategy, Usability, Question and Output Understanding, User Effort, Reliability, and Assessment Duration**.

Applying these criteria, we classified and evaluated 22 widely available FAIR-a tools. These tools were grouped into OSBSAT, OSAT, OAT, OFSBSAT, and OT, with results summarized in Tables 8.7, 8.8, 8.9, and 8.10. In the following, we describe tool evaluations within each group.

Online Survey-Based Self-Assessment Tools (OSBSAT): This group uses multiple-choice and radio inputs (sometimes with text boxes), focusing on dataset quality and individual knowledge. Assessments are generally easy to complete within 15 minutes.

SATIFYD is a user-friendly tool with drop-down and multi-select questions, featuring graphical outputs. Evaluation took 10.58 minutes. Results for the chosen dataset were Findable (88%), Accessible (100%), Interoperable (58%), Reusable (85%), and overall (83%). The tool was highly recommended for its ease of use and quantitative outputs.

Online Semi-Automated Tools (OSAT): These require both automated steps and user input, such as dataset DOIs or URLs.

FAIRshake was tested by creating a project named "SoBigData," selecting the repository type, and using an existing rubric. Despite spending 110 minutes to understand the tool and 20.73 minutes for evaluation, results were unclear. The tool is not recommended unless strictly necessary.

Online Automated Tools (OAT): This group includes nine tools requiring only links (PID, DOI, URL, URI).

F-UJI provided an excellent evaluation experience: the run took 1 minute 30 seconds, and output comprehension took 7 minutes 29 seconds, totaling 8.59 minutes. Outputs included a pie chart showing a FAIR score of 62%, with breakdowns across principles. We strongly recommend this tool.

Other tools in this group varied: some completed assessments within 15 minutes, while others required longer or suffered from documentation issues (e.g., FAIR Evaluator, OpenAIRE Validator).

Offline Survey-Based Self-Assessment Tools (OFSBSAT): These tools provide downloadable PDFs or Excel files for manual evaluation.

Do I-PASS for FAIR was tested using the editable PDF (22 questions in Policy, Services, Skills, Incentives, and Adoption). Evaluation took 22.05 minutes. Outputs were clear, but some questions were irrelevant for non-administrators. The tool was partially recommended.

Other Types (OT): This group includes tools offering awareness training through checklists and questionnaires.

FAIR-DSM (FAIRplus) comprises 17 questions across five maturity levels. The evaluation took 15 minutes and 22 seconds, with medium comprehension but clear

outputs. Supporting material from the FAIR Cookbook was included. Usability was medium, and we recommend this tool.

The comprehensive nature of our framework allows a detailed analysis of each tool's performance, ensuring the best and most time-efficient FAIR-a tools are identified. It reduces effort for selection, promotes best practices, and supports reproducibility and privacy in the health domain.

Answer to RQ2: The proposed evaluation framework is applied to systematically classify and compare 22 FAIR assessment tools. This application reveals significant differences in automation, usability, runtime variability, and interpretability. The results enable evidence-based tool selection and highlight key areas for improvement in the design and implementation of FAIR assessment tools within open science platforms.

8.5 Evaluation Result in selection of best FAIR-a Tools

This section presents the outcomes of our evaluation of 22 FAIR-a tools within the proposed framework. The goal is to identify the most effective tools from the user perspective, uncover limitations, and propose strategies for improvement. The results are organized around research questions RQ3 and RQ4, which together address the effectiveness of FAIR-a tools and the challenges associated with their use. Finally, we provide a broader discussion of the implications of these findings.

8.5.1 Addressing RQ3: Identify the best FAIR-a tools from the user perspective based on testing results.

After evaluating 22 FAIR-a tools, we assessed their quality and conducted a comparative analysis, organizing them into a table based on the original research framework. Through this framework, we were able to identify the overall better FAIR-a tools. In particular, 6 out of the 22 evaluated showed the highest level of user satisfaction. Moreover, we were able to identify the main challenges and limitations of using the tools.

For instance, we analyzed the FAIR-a tool **SATIFYD** by checking its attributes. *SATIFYD* is in a developed state and lacks adaptability but has a certain degree of maturity, providing quantitative results. It focuses on the four FAIR principles, emphasizes the FAIR Assessment strategy, and requires low effort to use. Based on user evaluations, *SATIFYD* is considered one of the best tools, though improvements such as increasing knowledge and verifying FAIR data are needed.

Building on this analysis, our framework identified the top six FAIR-a tools that excel in every dimension: **FAIR-Checker, FAIR enough [167], F-UJI, SATIFYD, FAIR-Aware, and FAIR Data Self-Assessment Tool**. These were categorized by specified bars, as shown in Figure 8.20. Additionally, there are seven tools that, while not the best in all actions, are still satisfactory and provide substantial value: **Do I-PASS for FAIR, HowFAIRIs, FOOPS!, NFDI4Culture FAIR-Check, FairDataBR, FAIRdat, and FAIR Data Self-Assessment Tool**.

Despite these advantages, several challenges and limitations exist when working with the tools in our framework. Understanding suitable data types for assessment and ensuring compatibility with different data formats is a primary concern. Some tools require standalone implementation, which can be challenging and complex.

Additionally, although guidance is provided on platforms like GitHub, individuals without extensive computer knowledge may struggle to understand the entire procedure.

Overall, our analysis demonstrates that existing FAIR-a tools are effective in measuring the FAIRness maturity of OSPs. At the same time, identifying their challenges highlights areas where improvements are needed, ensuring that these tools continue to evolve and better support the FAIR principles.

Answer to RQ3: The evaluation shows that a subset of FAIR assessment tools (*FAIR-Checker, FAIR enough, F-UJI, SATIFYD, FAIR-Aware, and FAIR Data Self-Assessment Tool*) demonstrates higher effectiveness in measuring FAIRness maturity. These tools achieve a balanced performance in terms of usability, runtime efficiency, and interpretability, making them suitable and reliable choices for assessing FAIR compliance within open science platforms.

8.5.2 Addressing RQ4: Discussing of FAIR-a Tool's detailed evaluation with analyzing results

The shortcomings of FAIR-a tools, including the lack of standardization, usability challenges, and limited coverage, can be addressed by improving automation, feedback systems, user-friendliness, and support for diverse data formats. To tackle these issues comprehensively, we propose a visual representation of FAIR evaluation results and actively gathering user feedback to better understand their concerns. Let's delve into the result analysis, discussing each figure step by step to gain deeper insights and connect them to the overall conclusions.

This pie chart in Figure 8.2 shows here the 22 selected FAIR-a tools, with online automated tools comprising 40.9% of the total, closely followed by online self-assessment tools at 31.8%. Offline self-assessment tools make up 9.1%, while the smallest group, online semi-automated tools, represents 4.5%. The graphic emphasizes the dominance of online automated tools in the dataset, indicating a clear preference for fully automated solutions within the FAIR-a tools landscape.

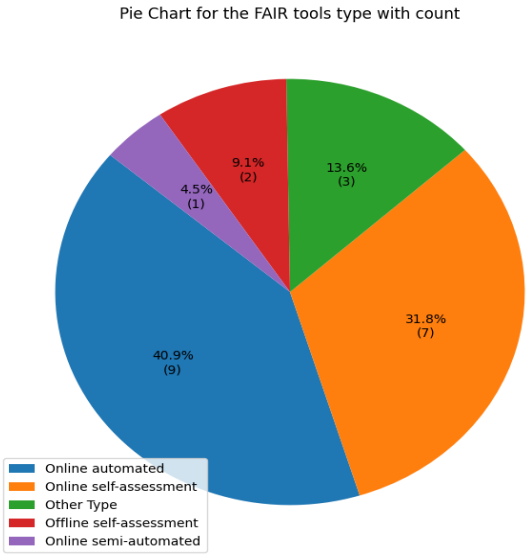


FIGURE 8.2: FAIR tool counts in each subcategory are shown in PieBar.

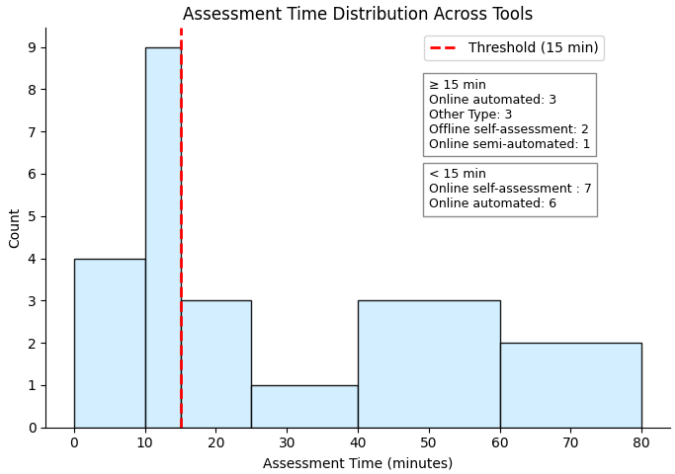


FIGURE 8.3: Assessment time distribution across tools. The dashed line indicates the 15-minute threshold used to distinguish faster and more time-intensive assessments, while annotations summarize tool counts below and above this limit.

The distribution of assessment times is shown in the histogram in Figure 8.3. The majority of evaluations are completed in less than 15 minutes, as indicated by the concentration of bars on the left side of the threshold marked by the red dashed line. In this range, online self-assessment and online automated tools are the most frequent, with 7 and 6 instances, respectively. Beyond the 15-minute threshold, fewer evaluations are observed, including online automated tools (3), other types (3), offline self-assessment (2), and online semi-automated tools (1). A few peaks can be observed at approximately 20, 49, 70, and 76 minutes, indicating isolated cases of longer assessment durations. Although a small number of tools require longer assessment times, extending up to around 80 minutes, the overall distribution shows that most tools can be evaluated quickly, with only a limited number requiring more time.

Figure 8.4 compares the distribution of FAIR-a tool types across two assessment time ranges defined by the 15-minute threshold. For shorter assessment durations ($<$ 15 minutes), the counts are dominated by online self-assessment and online automated tools, indicating that these tool types are generally quicker to evaluate. In contrast, for longer assessment durations ($\geq$ 15 minutes), the distribution becomes more diverse, with tools spread across multiple categories, including online automated, other types, offline self-assessment, and online semi-automated tools. This comparison highlights that while faster assessments are associated with a limited set of tool types, longer evaluations tend to involve a broader variety of approaches.

Figure 8.5 presents a violin plot showing assessment time in relation to tool applicability. The plot compares tools with a "yes limit" and "no limit" on dataset applicability. The central box represents the interquartile range (IQR) of assessment time, while the extended lines indicate the spread of values outside the IQR. The width of each violin reflects the concentration of assessment times at different values. Tools classified as having a "yes limit" tend to operate within constrained datasets, such as those limited by size or complexity, whereas "no limit" tools are designed to handle larger and more diverse datasets. Assessment time varies between these two categories, reflecting differences in how tools process datasets under applicability constraints.

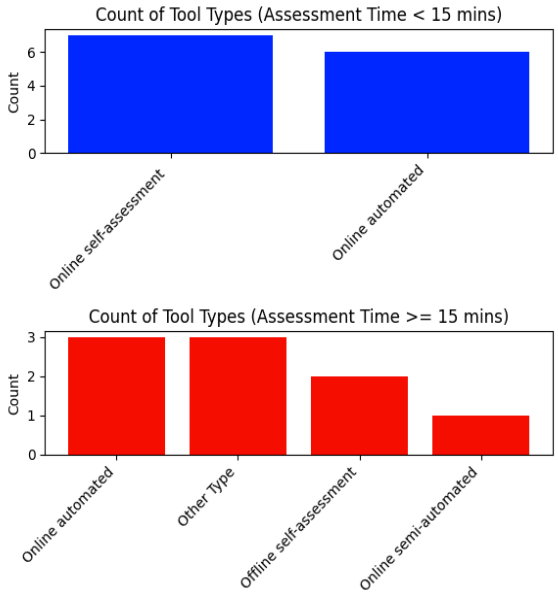


FIGURE 8.4: Blue Bars counted for *assessment time* $<$ 15 mins and *Reds* $\geq$ 15 mins.

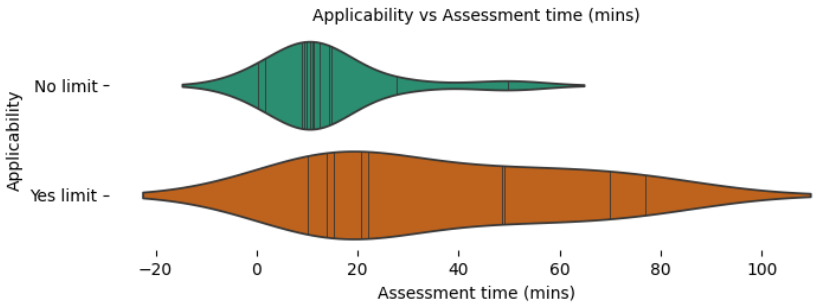


FIGURE 8.5: Assessment time vs. applicability constraints ("Yes limit" vs. "No limit"). Broadly applicable tools may exhibit higher runtime variability, suggesting scalability trade-offs.

The following Figure 8.6, which displays the level of understanding of the output concerning the tool's usability, comes after the time taken with the limitations of FAIR-a Tools. The following picture presents here a scatter plot that points out how user-friendly the FAIR-a tools are based on the kind of data input types. General users are advised to utilize the tools alongside the various input type options. Sometimes, the degree of tool usability determines the amount of understanding output; other times, it does not. Each data point in a scatter plot is represented by a marker on an x-y coordinate system, with one variable plotted along the x-axis and the other variable drawn along the y-axis.

The best input formats for assisting users in understanding the usability of a tool include multiple choice, radio, and web questionnaires, in addition to multiple choice to radio choices. The output of these tools is generally easy to understand, with some being medium to grasp. Both sides find multiple choice to be difficult. Certain URL kinds are easy to interpret the output, while others are medium. On both sides, the repository option is likewise moderate. Online questionnaires are a challenging tool to use, but it's easy to grasp the results. Some tools just accept,

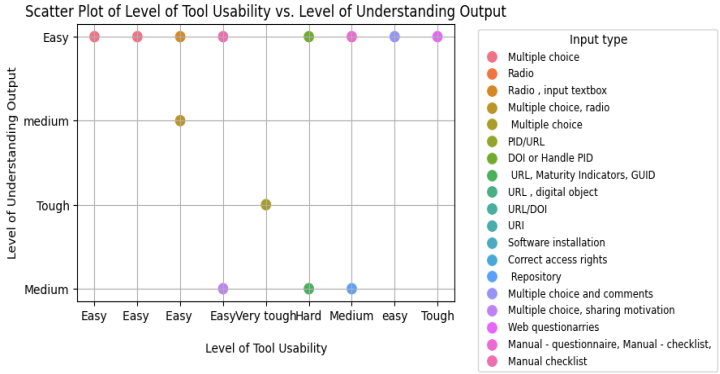


FIGURE 8.6: level of the FAIR tool's usability with understanding output in a scatter plot.

however, and it might be difficult to interpret the results.

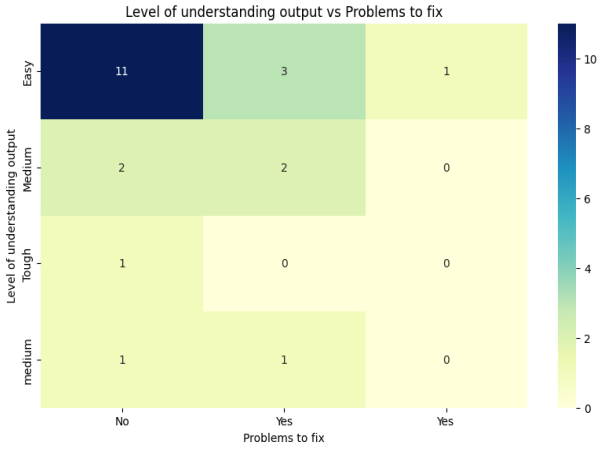


FIGURE 8.7: level of FAIR-a tool's understanding of the output and the problem to be fixed shown in a heatmap.

Therefore, as indicated in Figure 8.7, the necessity for repairs or improvements is influenced by the level of understanding of the output. This heatmap visualizes potential areas for future development or issues that need to be addressed based on feedback from our testing team. The majority of tools with understandable outputs have minor issues being fixed. Eleven tools, highlighted in the yellow box, are simple to use, with no challenges or required fixes. However, seven tools across different categories—three in the 'Easy' category, three in the 'Medium' category, and one in the 'Tough' category—have been identified for bug fixes or are flagged for future improvements. The tool development teams are focusing on these tools to address the issues raised by our testing team and to enhance their usability and performance.

Some of the FAIR-a tools are advised to be used, while others are not, based on the testing team's assessment of them. The histogram is used in Figure 8.8 below to display tool recommendations based on the type of input. It is highly advised to employ some of the resources listed in the manual checklist, such as web questionnaires, URL/PID, radio, and multiple-choice questions. We advise against selecting some radio, multiple-choice, and URI choices. The user needs to determine the software installation option as well. Sometimes users are also unsure whether to propose certain tools that are part of the multiple-choice selections. In order to

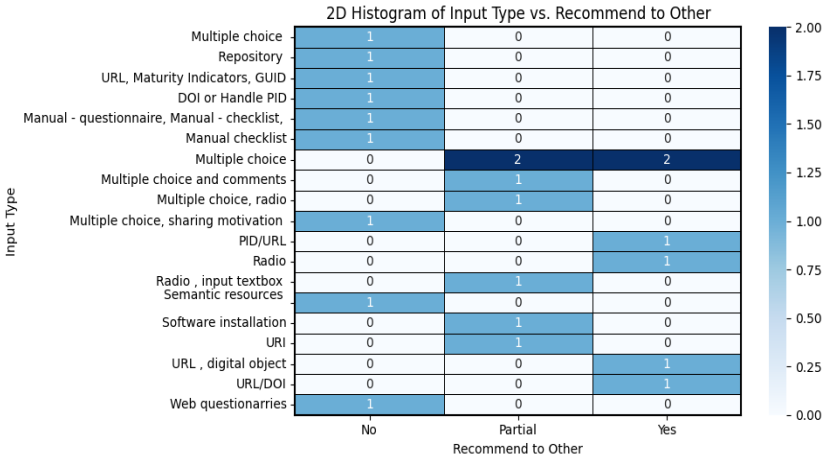


FIGURE 8.8: level of FAIR-a tool's understanding of the output and the problem to be fixed shown in histogram.

provide researchers with a brief discussion to aid in their decision-making process when selecting which FAIR tools to utilize, several figures are also included below.

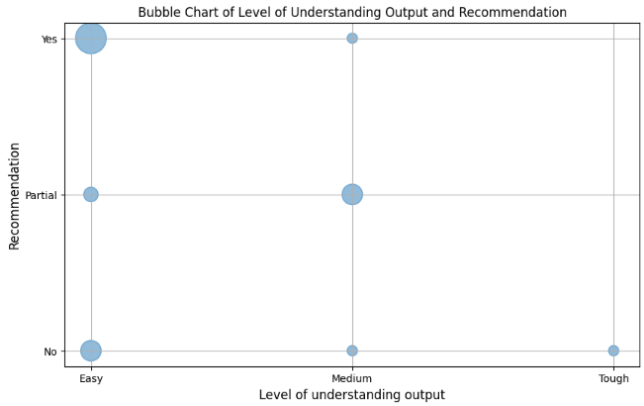


FIGURE 8.9: FAIR-a tool's recommendation with the understanding output shown in a bubble chart.

The understanding level of the output along with the suggestion from the testing team's experience is displayed in the bubble plot in Figure 8.9 above. Plotted on a two-dimensional axis, each bubble usually represents a single data point, and its location is dictated by the values of two variables. As can be seen in this figure, it is strongly advised to keep applying some of the tools with outputs that are simple for beginners to understand. While some are moderately easy to comprehend and come highly recommended, others rely on the specific needs of the user. Certain tools are extremely difficult to use and should only be used rarely.

An overview of the tool's recommendations based on the output degree of knowledge is provided by the violin plot below Figure 8.10. After evaluating twenty-two tools, we displayed the suggested tools in the figure based on the results of their analysis and knowledge level. This violin plot is broader in areas with higher density and narrower in areas with lower density, depending not only on the degree of user-friendliness but also on the level of understanding of the output. While several of the tools on this list are easily used, their understandability makes them unrecommended. While certain tools are difficult, they are advised to be used in future research.

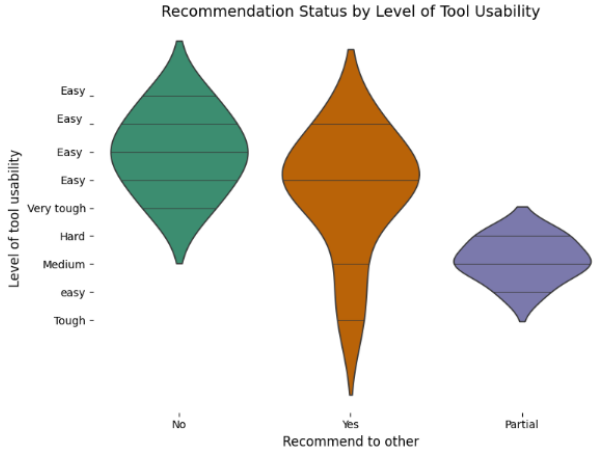


FIGURE 8.10: FAIR-a tool's recommendation with the level of tool's usability

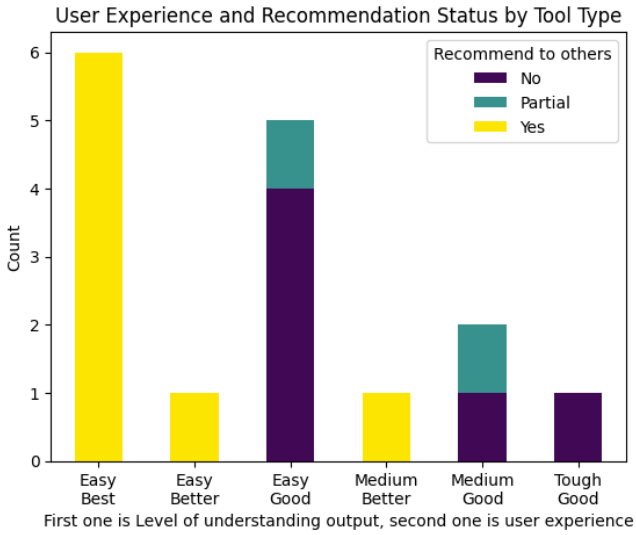


FIGURE 8.11: The first one is the level of understanding output, and the second one is our testing team's experience.

Our experience included things like tool usability, time required, and degree of understanding after completing the evaluation of 22 FAIR-a tools. The user in our team had the best, good, average, and bad experiences with these attributes. The degree of understanding of the results produced by FAIR technologies mostly determines these encounters. Therefore, it is very easy to understand which tools are examined depending on which kind in the figure that follows because of the nicely created bar plot. The x-axis of the bar graphs in Figure 8.11 displays the two factors that are combined: the team's experience and the level of understanding. On the y-axis is the number of tools represented. Concerning the categories of ['level of understanding output' and 'user's experience' as in team's experience], [Easy and Best] has the greatest count among the six tools. Moreover, it is not advised for some groups, such as [Medium, Good], [Tough, Good], and [Easy, Good]. For this group [Easy, Good], and for the [Medium, Good] group, the responses are 'Partial.'

8.5.3 Technical performance analysis about FAIR—a tool's

A few aspects are worth talking about before moving on to the analysis of the FAIR-a tools. The first of these is extensibility, or adaptability. The "yes" and "no" extensibility choices indicated if the tool was more or less strict in its usage and whether it was extendable (adaptable) for other users/developers. How developed the tool was may be determined by looking at its maturity level (yes/no options). The results type (qualitative/quantitative) follows these two attributes. Understanding the output for users is relatively simple and takes less time if the outcomes are quantitative. Additionally, if it is qualitative, users will need to invest more time in understanding, yet they will still struggle since they have never become used to the results. Next, the development state (developed or underdeveloped) is determined by the tools used, which rely on the FAIR principles (findable, accessible, interoperable, and reusable). Next, the tools' FAIR approach (Assessment / Improvement) focuses solely on assessment or improvement. Certain solutions just concentrate on evaluation and displaying outcomes; they do not address how to further enhance the data's FAIR quality. "Effort cost of using the tool" is another aspect that shows how simple the tool is to use and whether the developers have included any documentation that users need to read through before using the tool. The user-dependent evaluation grade is another significant aspect. In our testing team, the user then evaluated the tools in this case and rated them as "Best," "Better," or "Good" based on their performance.

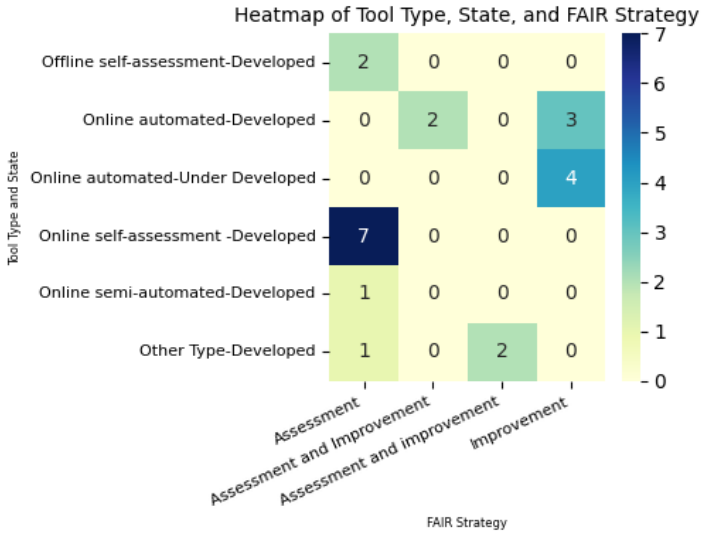


FIGURE 8.12: The HEAT map of analysis FAIR-a tool types with developing states and FAIR strategies

As seen in the below Figure 8.12, the heat map has three features: it groups states and tool types, and it displays the number of tools that use the FAIR strategy. This is a good summary that illustrates which tool types and how many tools are included in each FAIR method.

The above Figure 8.13 displays an exquisite line plot that describes each of the FAIR-a tool types using different colored lines. and the plot's x-axis has been structured according to targeted FAIR principles and FAIR states. Each group's tool count, represented on the x-axis, is represented by the peak points of every line in the figure.

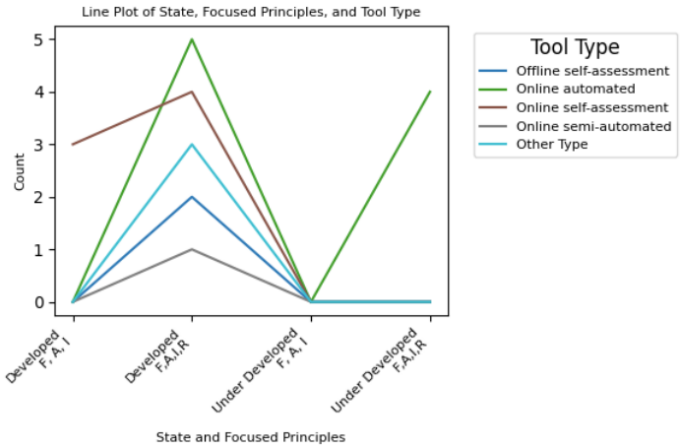


FIGURE 8.13: The Line plot of analysis FAIR-a tool types with developing states and focused principles

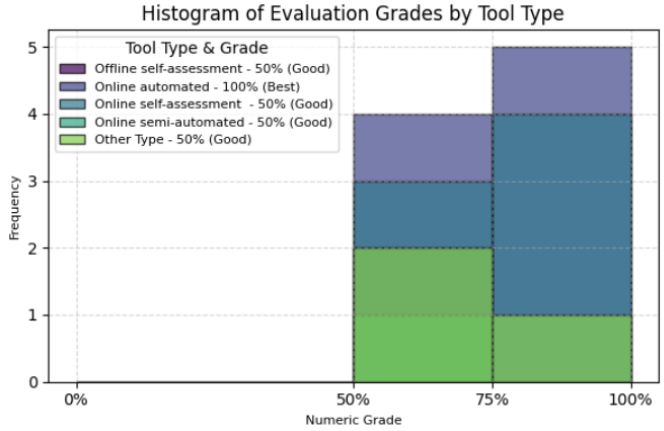


FIGURE 8.14: The Histogram of analysis FAIR tool types with Evaluation grades

The frequency of evaluation ratings is displayed in the histogram in Figure 8.14 above, where 100% represents Best, 75% represents Better, and 50% represents Good. Tool types are displayed in distinct colored boxes. This suggests that while many tools perform adequately, online automated tools are considered the most effective, highlighting their superior performance in the evaluation.

The outcome of the tool types in each evaluation grade is clearly displayed in the grouped bar chart, Figure 8.15. Three distinct bars, each with a different hue heaped inside, show the number of tool types and their corresponding percentages. This figure helps guide researchers in determining which group of FAIR-a tools are most suitable for further use in their research. By showing the distribution of evaluation grades, it highlights that both online automated and online self-assessment tools are rated higher, making them the best choices compared to other groups. While these two types of tools are top-performing, other tool types generally perform well but may not reach the same level of effectiveness, offering researchers a clear comparison to inform their decisions.

Another good scatter plot showing the effort put in by each tool type according to the evaluation grade is the following Figure 8.16. Here, the tool type and effort are used to indicate the assessment grades of "Best," "Better," and "Good," which are represented by each color.

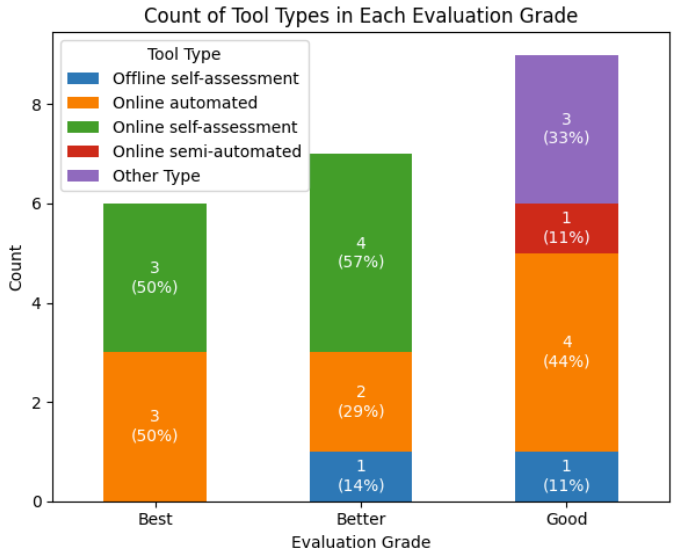


FIGURE 8.15: The Grouped BAR chart of analysis FAIR-a tool types (count and percentage in each grading) with Evaluation grading

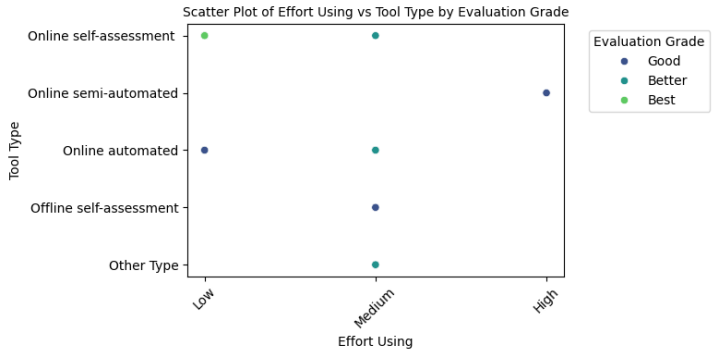


FIGURE 8.16: The Scatter plot of Effort using in each Tool type and grouping by Evaluation grade

The heat map in Figure 8.17 below is the outcome of the tool names that are included in the development states. The grade mapping is shown by the numbers inside the mapping, which are "Best" at 1, "Better" at 2, and "Good" at 3. It is evident from this figure that the four best tools for evaluation grading are those from the "Developed" state and the two from the "Under Developed" state.

Along with the tool names, the count is displayed on the x-axis in the bar plot with several evaluation grades. The best grade of the sky blue-colored bar is clearly occupied by six tools in this Figure 8.18, which is highly recommended from the user's perspective. Additionally, seven tools fall under the 'Better' evaluation grade, which is also recommended, though partially. The remaining tools under the 'Good' grade are usable but are not as strongly recommended based on the user's evaluation.

The study of six different tool types is displayed in the following heat map 8.19. and the FAIR strategy is also indicated inside the heat map. The six tools for best evaluation that are taken into consideration concerning USER1's experience are "F-UJI", "FAIR data Self-Assessment Tool", "FAIR-Aware", "FAIR-Checker", "FAIR enough" and "SATIFYD". In the same way as the following line and scatter plot in Figure 8.20, ten tools ("FairDataBR", "FOOPS!", "FaIRdat", "F-UJI", "FAIR-Aware", "FAIR-Checker",

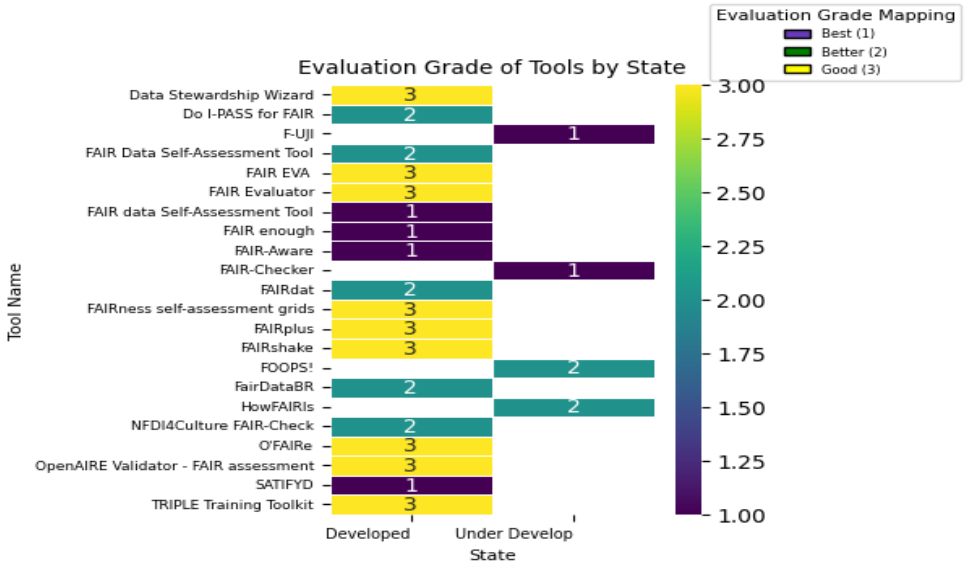


FIGURE 8.17: The HEAT map of FAIR—a tool named for developing states and Evaluation grading

FIGURE 8.18: The Bar plot of Evaluation grade with count of Tools Name

"SATIFYD", "FAIR Data Self-Assessment Tool", "FAIR enough", and "FAIR data Self-Assessment Tool") are considered to require less effort to use, with the types of tools indicated on the x-axis and the grades "Better" and "Best" based on these grades.

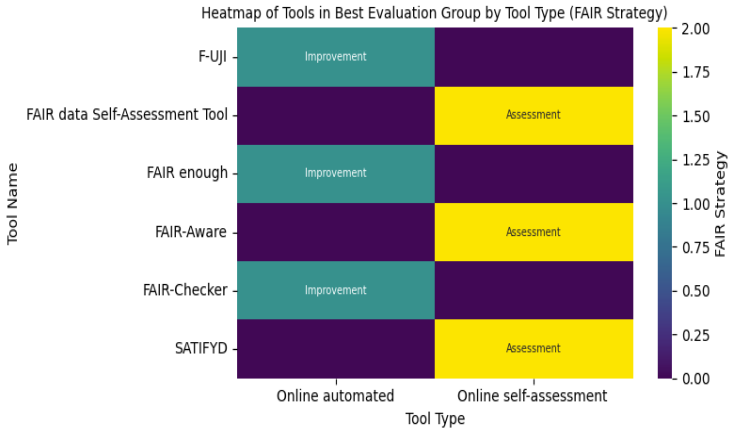


FIGURE 8.19: Heatmap of Tool Names in Best Evaluation Group by Tool Type and FAIR Strategy

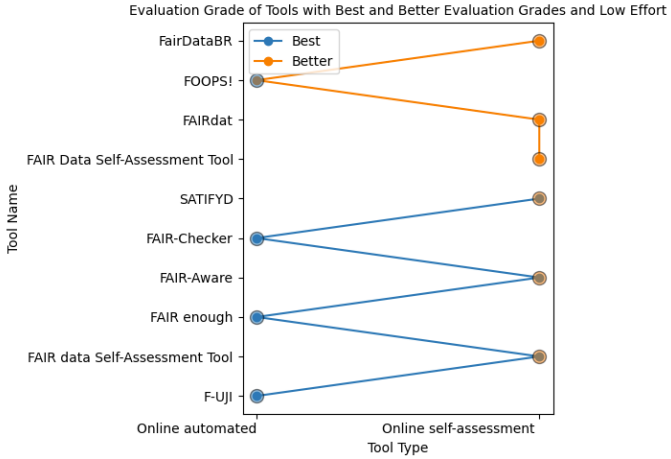


FIGURE 8.20: Evaluation Grade of Tools with Best and Better Evaluation Grades and Low in Effort using the tool

Answer to RQ4: The evaluation reveals several recurring limitations in FAIR assessment tools, including performance variability (particularly execution time), unclear or non-actionable assessment outputs, usability barriers related to input requirements, and limited applicability across datasets, metadata formats, and repository types. To enhance usability and reliability, future tools should provide clearer scoring explanations, more actionable feedback, simplified user interactions, and broader support for diverse data objects and repository environments.

8.6 Discussion

8.6.1 Recommendations Based on the Analysis of FAIR Tools' Performance

After several discussions, the plot results are finally ready for evaluation. It's clear that while some tools are straightforward and user-friendly, others require significant effort to understand their output. In such cases, those tools may not be suitable for general use, especially by users who are not technically experienced. Certain tools are highly suggested in circumstances when they have the best expertise and

produce easily understood results. Each group in the below Figure 8.21 has a large number of tools, and it indicates which groupings are suggested and which are not.

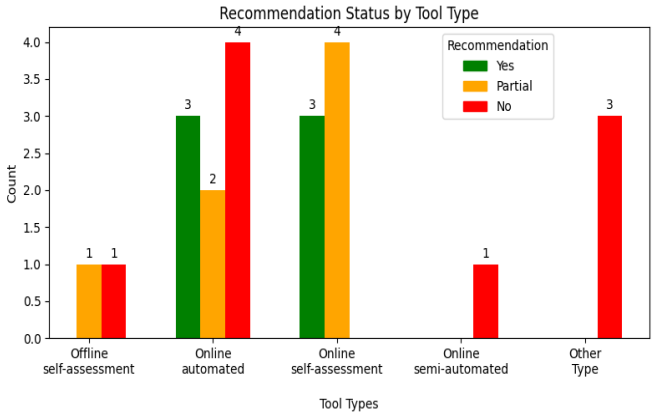


FIGURE 8.21: Grouped bar plot showing the number of recommended FAIR-a tools across different tool groups.

Some tool development teams also mentioned that their tools are still under development and that they expect future versions to deliver more accurate and reliable results.

The following section discusses the specifics of the FAIR-a tools. Each group type is described in detail in Figure 8.22. The assessment in our team serves as the only ground for the recommendation. We will also keep working on the evaluation of more users using the same tool with several data. Accordingly, the figure indicates that for the first group's online survey-based self-assessment, three tools have been recommended for use: *SATIFYD, FAIR-Aware, and the FAIR data self-assessment tool (Cross-Lateral Enterprises Pty Ltd)*.

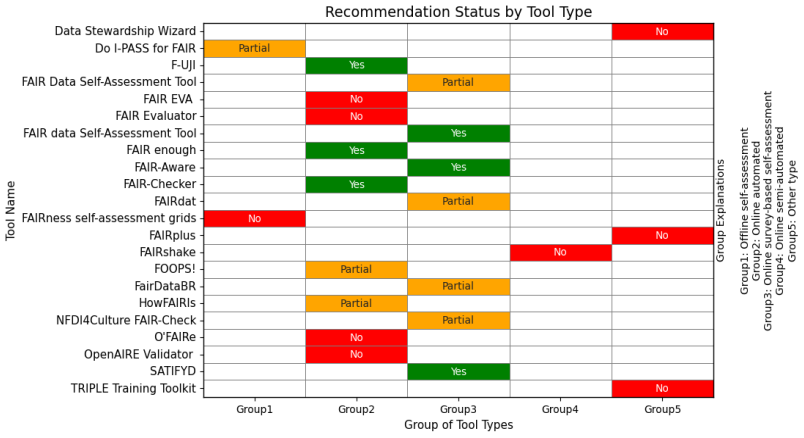


FIGURE 8.22: List of recommended FAIR-a tools along with their respective tool group categories.

Additionally, in the automated online group, it is suggested to use three tools because they are easy to use, take less time, and help users grasp the output better overall. These tools had high scores. These tools are *FAIR-Checker, F-UJI, and FAIR enough*. Additionally, the usage of additional technologies is advised since they involve large businesses and the requirement for more team members to share expertise. The only tool utilized in this online semi-automated group evaluation

is FAIRshake, which is difficult enough for first-level users to use and is thus not advised.

Despite the fact that the evaluation framework enables a systematic comparison among FAIR evaluation tools, there are some limitations that need to be addressed. The behavior of the tools could differ depending on the software version, which could influence the reproducibility of the scoring results. Moreover, there were some evaluation criteria that needed to be interpreted manually from the tool results, which could introduce some subjectivity despite the standardized scoring rules. In addition, the evaluation is mainly based on technical aspects and does not necessarily cover the domain-specific FAIR implementation details.

8.6.2 Threats to validity

This section explains the main factors that may affect the reliability of the results from the FAIR-a tool's evaluation. Each type of threat is described separately for clarity.

- **Internal validity.** In the FAIR-a tool evaluation, internal threats mainly come from human judgment while interpreting the tool outputs. This was handled by using fixed evaluation criteria and testing all tools on the same dataset (MMASH).
- **External validity.**

  In FAIR-a tools, external threats appear because results may change when different dataset types or more users are involved. Although one dataset was used for consistency, future testing on more datasets will help confirm the findings.
- **Conclusion validity.**

  For FAIR-a tools, the conclusions are based on the 22 tools selected for review. New tools may appear later, and comparing them again will be necessary to keep the findings valid.
- **Baseline validity.**

  In the FAIR-a tools study, baseline validity refers to how manual, semi-automated, and automated tools were compared. While the evaluation was kept as uniform as possible, more testing across updated tools will improve consistency.
- **Statistical validity.**

  In the FAIR-a tools evaluation, most observations were qualitative. Adding numerical measures such as execution time or user ratings in future work will provide stronger statistical support.
- **Preprocessing validity.**

  For FAIR-a tools, preprocessing validity relates to the completeness and structure of the dataset metadata. If the metadata changes, the FAIR scores may also change. Using more standardized metadata will help reduce this issue.

Overall, these threats highlight the need for standard procedures, clear reporting, and repeated experiments in this part of the thesis. Even with these limitations, the results remain stable within the given scope and provide a useful base for future work in FAIR data assessment.

8.7 Conclusion

This research work provides a comprehensive assessment of various FAIR-a tools, focusing particularly on the Multilevel Monitoring of Activity and Sleep in Healthy People (MMASH) dataset. The study classifies these tools into five distinct groups and highlights the general relevance of the results across different research domains. The classification includes quick online survey-based self-evaluation tools, semi-automated processes, fully automated online tools, offline self-assessment templates, and tools primarily used for project data management plans or training purposes.

From this classification, six specific FAIR-a tools have been identified as highly useful for researchers. For rapid online self-assessment, 'FAIR-Aware,' 'FAIR data Self-Assessment Tool,' and 'SATIFYD' are recommended. For automated online evaluations, 'F-UJI,' 'FAIR enough,' and 'FAIR-Checker' are suggested as shown in Figure 8.22.

The usability of these tools is a key consideration, especially for first-time users who may be unfamiliar with the FAIR principles. Out of the thirteen recommended tools, six stand out for their ease of use and high evaluation grades and have been distinguished with the best evaluation grades, categorizing them as either "Online automated" tools or "online survey-based self-assessment" tools, as shown in Figure 8.19. These tools are particularly suitable for new users, who can rely on the provided materials and tool descriptions to effectively assess and improve the FAIRness of their data.

In conclusion, this study highlights the importance of selecting the right FAIR-a tools to enhance the efficiency and impact of research within Open Science Platforms (OSPs). By recommending tools based on usability and evaluation grades, it supports the broader adoption of FAIR principles, fostering a more transparent and accessible research environment. Future efforts should validate these findings and develop new tools to address existing limitations, ensuring ongoing improvements in data management practices.

In our future work, we plan to involve a wider group of researchers in the assessment and evaluation process to help validate the framework on a global level. By encouraging broader participation, we hope to establish this framework as a valuable reference point for early-career researchers working in the field of open science. Additionally, our development team is motivated to build a new FAIR-a tool that addresses existing inefficiencies. This tool will be designed to be highly user-friendly, providing accurate results with a precise FAIR score. By enhancing usability and accuracy, we aim to further promote the adoption of FAIR principles and contribute to the advancement of open science.

Chapter 9

Discussion and Interpretation of Results

This chapter addresses the study's main outcomes and considers their broader implications. The outcomes from the several experimental components—clinical entity extraction, RLGMO-based therapy optimization, and EAT (Epicardial Adipose Tissue) quantification—are analyzed in Section 9.1 to comprehend their technological behavior, advantages, and disadvantages. After that, Section 9.2 offers a general discussion that links these results, emphasizes the key findings from each study, and considers how they all work together to support the thesis's goals. The practical difficulties, methodological revelations, and chances for additional improvement that surfaced during the study are also taken into account in this chapter.

9.1 Interpretation of Results

9.1.1 Medical Data Analytics and AI-driven Framework

The results of this research study demonstrate that integrating artificial intelligence, reinforcement learning, and knowledge graph modeling provides a robust foundation for medical data analysis and clinical decision support. The proposed pipeline successfully processed unstructured clinical text and transformed it into structured, interpretable data. The following key outcomes were observed:

1. **Accurate Entity Extraction:** The AI-driven entity extraction pipeline efficiently identified relevant medical entities such as *diseases, symptoms, medications, doses, and cancer stages* from unstructured clinical notes, demonstrating the capability of fine-tuned language models to capture domain-specific terminology.

2. **Knowledge Graph Representation:** The extracted entities were effectively represented in a structured knowledge graph format, allowing the relationships between diseases, symptoms, and medications to be visualized and semantically linked. This enabled improved interpretability and data reusability for downstream clinical tasks.

3. **Reinforcement Learning and Graph Integration:** The proposed **RLGMO (Reinforcement Learning Guided Medicine Optimization)** in the MetaGRO framework demonstrated that reinforcement learning combined with graph neural representations can yield adaptive and context-aware therapeutic recommendations. This hybrid approach enhanced personalization and interpretability in oncology decision-making.

4. **Improved Transparency and Reproducibility:** The use of knowledge graphs makes the decision-making process traceable and understandable. Each recommendation can be linked back to the underlying data and reasoning path, ensuring that results are consistent, interpretable, and reproducible across different datasets. This structure supports the FAIR principles by promoting data clarity, accessibility, and reuse.

Collectively, these results show that combining AI and reinforcement learning with graph modeling can improve both the precision and trustworthiness of clinical recommendations, setting a foundation for future FAIR-aligned clinical decision support systems.

9.1.2 Evaluation of FAIR Assessment Tools

The second major outcome of this thesis focused on the evaluation and classification of existing FAIR-assessment (FAIR-a) tools. The study involved a systematic comparison of tools based on their automation level, usability, and reliability, leading to several important interpretations:

1. **Variation in Tool Performance:** The evaluation results showed that different FAIR-assessment (FAIR-a) tools vary widely in how efficiently they measure FAIRness. Some tools are simple and quick to use but lack accuracy, while others provide detailed evaluations but require technical expertise. This shows that no single tool currently provides a complete and balanced FAIR assessment experience.

2. **Grouping for Better Understanding:** To make the comparison more meaningful, the tools were organized into four main groups: *online survey-based*, *offline survey-based*, *semi-automated*, and *fully automated*. This classification helped highlight specific strengths and weaknesses—for instance, survey-based tools are easier for beginners, while automated ones are faster but often less flexible in handling complex datasets.

3. **Need for a Unified Platform:** The results revealed a clear gap: researchers need a single platform that can combine automation with ease of use. Current tools either require too much manual effort or fail to give clear and user-friendly feedback. The second part of this study resulted in the development of a novel integrated framework for FAIR assessment.

4. **FAIR-IS-ON as the Next Step:** To fill this gap, the **FAIR-IS-ON** tool is being developed as a semi-automated, intuitive platform that brings together the best features of existing tools. It aims to provide accurate FAIR scoring, clear visual feedback, and flexibility for different datasets, ensuring that users can easily evaluate and improve their data's FAIR compliance.

Together, these interpretations underscore the complementary nature of the two research dimensions—AI-driven medical analytics and FAIR data evaluation. Both converge toward a shared goal of enabling transparent, accessible, and reproducible research ecosystems that align with the vision of open science.

9.1.3 Echocardiographic EAT Quantification

The additional experimental study on *Echocardiographic Epicardial Adipose Tissue (EAT) Quantification* explored the feasibility of automating EAT thickness measurement using standard transthoracic echocardiogram videos. This work, presented at the `PerFail` 2024 Workshop on Negative Results in Pervasive Computing, aimed to identify the technical challenges involved in extracting clinically meaningful EAT measurements from low-contrast and noise-prone echocardiographic images. Based on the dataset of 152 echocardiogram videos and a multistage image processing pipeline, several important findings emerged:

1. **Image Quality and Noise Sensitivity:** The echocardiogram frames were highly affected by speckle noise, motion artifacts, and varying anatomical orientations. Despite extensive preprocessing—including masking, cropping, noise reduction, and binary conversion—only 23 out of 152 frames met the minimum visual clarity for analysis. This drastically reduced dataset highlighted the difficulty of obtaining consistent EAT visibility in routine clinical imaging.

2. **Failure of Classical Line and Edge Detectors:** Multiple established image processing algorithms, such as Sobel edge detection [168], Harris and Shi–Thomasi corner detectors [169], [170], Hough Transform [171], Parallel Line Detector [172], and Best-Fit line estimation [173], were applied to identify the two hyperechoic boundaries that define EAT thickness. However, none of these methods consistently detected the correct upper and lower EAT borders due to pixel noise, irregular tissue patterns, and lack of strong linear structures in the images.

3. **Challenges in Frame Extraction and Standardization:** The assumption that extracting one frame per second would reliably capture the telesystolic moment did not hold true. Patient-specific cardiac timing variations made it difficult to isolate the correct end-systolic frame, often resulting in inconsistent EAT visibility. This emphasized the need for more robust frame-selection strategies or machine-learning-based cardiac phase identification.

4. **Limitations in Segmentation Feasibility:** The masking and focused cropping steps, though intended to simplify analysis, removed essential contextual information required by line-based algorithms. Without clear structural cues, segmentation methods frequently misinterpreted other bright anatomical areas as EAT, leading to unstable output.

5. **Need for Advanced Learning-Based Approaches:** The study demonstrated that purely traditional image-processing workflows are insufficient for reliable EAT quantification in echocardiography. Deep learning–based models trained on expert-annotated datasets or clustering-based segmentation techniques may offer better potential for capturing the complex, irregular boundaries of EAT.

Overall, our experimental study highlighted the underlying challenge of automating EAT measurement using echocardiographic videos. The pipeline underlined that more sophisticated, annotation-driven, and learning-based techniques are required to obtain clinically reliable EAT quantification, even though it provided helpful insights into preprocessing, border detection, and variability in cardiac imaging quality.

9.2 Overall Discussion and Reflection

This thesis demonstrated the potential of integrating artificial intelligence, reinforcement learning, and FAIR principles to improve medical data processing, clinical decision support, and open science practices. The proposed framework successfully converted unstructured clinical text into structured and interpretable information by extracting key entities such as diseases, symptoms, medications, and cancer stages. Fine-tuned transformer models, especially BioBERT, provided the most reliable results and formed a strong baseline for practical use in breast cancer information systems.

The consistency observed across model evaluations highlights the robustness of the fine-tuning approach and confirms the value of domain-specific language modelling for biomedical data. The structured information obtained from entity extraction supports the development of a Medical Knowledge Graph (MKG) that links related clinical concepts and helps identify useful patterns across patient data. Building on this foundation, the MetaGRO framework adds a graph-based optimization layer that uses reinforcement learning to explore treatment options, identify clinically relevant relationships, and support personalized decision-making. Together, the MKG and RLGMO components show how structured biomedical knowledge can strengthen precision medicine through clearer reasoning and adaptive recommendations.

Because the MetaGRO framework operates as a sequential pipeline, the reinforcement learning module directly depends on the quality of the structured entities extracted in earlier stages. As such, any level of uncertainty that was introduced in the earlier stages of entity extraction could potentially impact the knowledge representation and the reinforcement learning optimization process.

9.2.1 System-Level Integration of FAIR Principles in MetaGRO

This thesis also considers the implications of FAIR principles within the architectural design of the MetaGRO framework. While FAIR is traditionally discussed in the context of data governance and repository deposition, this work emphasizes its relevance at the system level in AI-driven medical analytics. In this context, FAIR-based data governance and AI-based medical analytics are not only sequentially integrated but also conceptually connected. Medical ontologies such as UMLS, SNOMED CT, ICD, and RxNorm play a central role in the construction of the knowledge graph, as they promote semantic interoperability and structured knowledge reuse within healthcare systems. By linking extracted entities to these standardized vocabularies, the framework improves consistency, interpretability, and interoperability across heterogeneous medical data sources. Within the MetaGRO framework, each RLGMO instance is represented as a structured and typed subgraph that explicitly connects patient attributes, diagnoses, symptoms, medications, and relevant clinical guidelines. This graph-based representation enhances machine readability, improves traceability of clinical relationships, and supports more stable and interpretable reinforcement learning optimization.

At present, FAIR principles are primarily operationalized at the dissemination and governance levels through the structured storage of datasets, embeddings, and trained models within open science platforms. However, the architectural design anticipates deeper FAIR-by-design integration at the system level. Future work will explore mechanisms for embedding provenance tracking, uncertainty metadata propagation, and model lineage documentation directly within graph nodes

and edges. Such extensions would further strengthen the alignment between AI reasoning processes and FAIR data stewardship practices. Taken together, these architectural considerations showcase how FAIR principles can extend beyond traditional data governance and become structurally embedded within AI-driven medical reasoning systems. Beyond the architectural perspective, this work contributes to a broader understanding of how medical data can be made FAIR, ensuring that structured and machine-readable information supports transparency, interoperability, and reproducibility. Although the primary focus of this study was clinical entity extraction, the same methodological framework can be extended to FAIR-compliant, AI-assisted Clinical Decision Support Systems (CDSS) that operate within ethical and privacy-preserving healthcare environments.

Despite certain limitations—such as the limited dataset size, single-run experiments, and variations in preprocessing—the results remain consistent within the scope of the study. The findings suggest that combining biomedical language models, knowledge graph construction, and reinforcement learning is both feasible and effective when aligned with FAIR principles. Overall, this research connects advanced AI techniques with practical clinical needs and provides a foundation for future healthcare systems that are scalable, transparent, and ethically responsible.

Takeaway:

- The *Clinical Entity Extraction* work has resulted in a journal paper titled *"Entity Extraction in Clinical Summary of Mammary Malignancy Data using Transformer-Based Models,"* which has been submitted to **Applied Soft Computing**. This study focused on extracting important clinical information from mammary malignancy summaries using transformer-based NLP models such as BioBERT, ClinicalBERT, and other domain-adapted architectures.

  The experiments showed that these models performed much better than traditional machine learning methods, especially in recognizing medical terms that appear as multi-word phrases or require contextual understanding. However, at the same time, challenges are also identified by the investigation. Incomplete clinical descriptions, improvised clinical words, and unusual medical language that are frequently included in actual clinical notes were difficult for the models to understand.

  Overall, this study highlights the strong potential of transformer-based models for clinical entity extraction. However, achieving dependable and clinically meaningful performance will require richer annotated datasets, careful model tuning, and comprehensive evaluation across larger and more diverse datasets beyond the current experimental setting.

- The *RLGMO* study, another module of the MetaGRO Framework, explored how reinforcement learning can be combined with medical entity knowledge graphs and graph neural networks to support therapy optimization. This work is being prepared for submission to the journal **Artificial Intelligence in Medicine** under the title *"Integrating Medical Entity Knowledge Graphs with GNN-Based Pattern Analysis for Reinforcement Learning-Based Therapy Optimization (RLGMO)."*

The experiments showed that incorporating structured medical knowledge helped the reinforcement learning system produce more consistent and clinically aligned decisions. The use of knowledge graphs and GNNs (Graph Neural Networks) allowed the model to better understand the relationships between medical entities (Disease–Symptoms–Medication), which reduced contradictory or incoherent therapy suggestions. However, the study also highlighted some significant issues. These include the additional computing burden in processing graphs, the narrow scope of existing medical ontologies, and the challenge of creating a reward mechanism that effectively directs the reinforcement learning model.

Overall, the study pointed out that combining knowledge-graph-based representations with reinforcement learning can aid in making therapeutic decisions that are easier to understand and have greater clinical significance. However, better reward-shaping techniques, more complex and comprehensive ontological resources, and broader evaluation beyond the current experimental setup are required for this strategy to become more reliable.

- As an extended contribution, a separate experimental study on *Echocardiographic Epicardial Adipose Tissue (EAT) Quantification* was carried out and presented in the ***PerFail 2024 Workshop*** on Negative Results in Pervasive Computing. The study aimed to automate the quantification of EAT thickness from echocardiographic video frames to assist cardiologists in cardiovascular risk assessment and monitoring. The proposed workflow consisted of five key stages, starting from video preprocessing to the detection of the upper and lower EAT boundaries. Several traditional computer vision methods—Sobel Edge Detection, Harris Corner Detection, Hough Transform, Shi–Thomasi Corner Detection, Parallel Line Detection, and Best-Fit Algorithm—were evaluated and compared for their ability to identify the EAT contour.

  However, due to the inherently low contrast, high noise level, and irregular structure of echocardiographic images, these conventional algorithms produced inconsistent and unstable results. The edge detectors struggled to isolate the true EAT boundary from surrounding cardiac tissue, while line detectors identified thousands of irrelevant segments caused by noise and nonlinear tissue geometry. These findings highlighted that the quantification of EAT thickness remains a complex and open research problem, particularly when using 2D echocardiography data without standardized imaging conditions.

  To overcome these limitations, a clustering-based algorithm was later introduced to detect and group white pixel regions corresponding to the upper and lower EAT layers. This approach allowed preliminary estimation of EAT area and thickness through region clustering and pixel-based measurements. Although the clustering stage remains under development, it demonstrated promising potential for improving detection accuracy and computational efficiency.

  As a next step, the study envisions extending this methodology through deep learning techniques. By training neural networks on annotated echocardiographic datasets, it will be possible to simultaneously segment and measure EAT regions with higher precision and generalizability. Such models could

support cardiologists in large-scale clinical screening, enhance the reproducibility of measurements, and deepen the understanding of the relationship between EAT and cardiovascular diseases (CVDs).

In conclusion, our study demonstrated the difficulty of evaluating echocardiographic pictures and the significant limits of conventional segmentation techniques. These results show that accurate quantification of epicardial adipose tissue (EAT) is achievable only by repeated testing, algorithm transparency, and continuous methodological improvement because of the complexity of echocardiographic data.

Chapter 10

Conclusion, Lesson Learned, and Future Directions

This chapter summarizes the thesis's overall contributions with lessons learned and suggests further research directions. The main lessons learned are discussed in Section 10.1, emphasizing the methodological thoughts and useful insights obtained from this study. The main contributions made by the various research components are summarized in Section 10.2, emphasizing how each component of the study promotes the knowledge and use of FAIR data practices and AI-driven medical data analysis. Subsequently, Section 10.3 outlines possible directions for further research studies, highlighting improvements, expansions, and novel research routes that can expand upon the results of this study. When combined, these parts provide a clear overview of the thesis and a perspective for further developments in this field.

10.1 Lessons Learned

Over the span of this work, we discovered several meaningful lessons that shaped our technical approach and clarified the role of AI and FAIR principles in improving healthcare data management

- **Clean and well-structured data is essential.** Working with unstructured clinical text showed that high-quality, well-annotated data is crucial for accurate analysis. Converting raw clinical notes into structured formats improved model performance and made the extracted information easier to interpret and reuse.
- **Domain-specific fine-tuning is necessary for medical language tasks.** The experiments demonstrated that fine-tuning biomedical language models such as BioBERT leads to significantly better results than relying on general pretrained models. Fine-tuned models were better at recognizing medical terms and their meanings than general models.
- **Enhancing clinical impact through the integration of AI with knowledge-based techniques.** By combining entity extraction with knowledge graphs and reinforcement learning, we were able to build a framework, MetaGRO, in a way that allows users to understand its decisions and modify its behavior. The RLGMO model in this framework showed that this method can pick up meaningful clinical patterns and help with personalized treatment choices.
- **User-friendly tools are more likely to be adopted.** The FAIR-a tool evaluation's findings highlighted that effective adoption requires more than just accuracy. Users want tools that are simple to use and perceive. The FAIR-IS-ON

platform, which attempts to simplify FAIR evaluations while ensuring reliability, was designed with this idea in mind.

- **Ethical, transparent, and reproducible AI is essential.** Protecting data, following ethical rules, and keeping experiments reproducible were as important as the results. Applying FAIR principles makes the AI work clearer and more useful for researchers and healthcare workers.

To summarize, this work showed that strong technical results alone are not enough in healthcare AI. Data quality, usability, ethics, and transparent reporting are equally important. These aspects help in creating reliable and FAIR-aligned systems for future medical applications.

10.2 Summary of Contributions

This thesis presents a combined research framework that brings together artificial intelligence and reinforcement learning with clinical data following the FAIR data principles to support medical data analysis and open science practices. The work is organized into two main parts.

The first part focuses on medical data analysis and clinical entity extraction. A transformer-based pipeline was developed using pretrained biomedical BERT models such as BioBERT, BioClinicalBERT, PubMedBERT, BlueBERT, and the general RoBERTa model to extract key entities—*Age, Gender, Disease, Symptoms, Medication, Dose, and Cancer Stage*—from unstructured clinical text. Among these models, fine-tuned BioBERT gave the most reliable results. The structured data produced from this first step provided the basis for building Medical Knowledge Graphs (MKGs) that represent the links between diseases, symptoms, and medications. Such graphs can support personalized clinical decision-making.

The second part of the research evaluates the existing FAIR-assessment (FAIR-a) tools and introduces a new FAIR-oriented framework. A comparative study of 22 tools was carried out, grouping them into survey-based, semi-automated, and fully automated categories. This analysis showed that there is a need for a more integrated and easy-to-use system. This observation guided the development of the **FAIR-IS-ON** tool, a semi-automated platform designed to support FAIR evaluation.

Together, these contributions show how AI-based data analysis and the concept of FAIR principles can be brought together to improve transparency, reproducibility, and accessibility in healthcare and medical research settings. The methods developed in this thesis provided a useful starting point for building FAIR-aligned, explainable, and trustworthy healthcare systems.

10.3 Future Research Directions

Future research will focus on strengthening both parts of this research study so that it can make a clearer clinical contribution and better support FAIR principles.

- **Enhancing Medical Knowledge Graphs.** Our future work will concentrate on expanding the Medical Knowledge Graph (MKG) by incorporating other data sources, such as external medical ontologies and long-term patient records. We intend to investigate advanced GNN architectures, dynamic graph updates,

and broader clinical validation to further improve treatment suggestion quality and promote individualized care, building on the current application of Graph Neural Networks (GNNs) and reinforcement learning inside the RL-GMO module.

- **Development of the FAIR-IS-ON Tool.** The FAIR-IS-ON tool is being developed as a semi-automated web platform for FAIR assessment. Further versions will include API support, visual dashboards, and backend FAIR metric checks to increase automation and interoperability across datasets.
- **Increasing Dataset Diversity.** The present work uses an unannotated dataset on breast cancer and pancreatic cancer. Expanding the study to larger datasets from other medical areas—such as cardiology and neurology—will help test the generalizability of the approach and show how well it scales to different domains.
- **Improving Experimental Reliability.** Future experiments will be run multiple times with fine-tuning with different random seeds, and the mean and variance will be reported to ensure statistical reliability. The impact of preprocessing on clinical entity extraction performance will be better understood with additional comparisons between filtered and unfiltered text, assessed against the same ground truth annotations.
- **Integration with Clinical Decision Support Systems.** Our long-term goal is to integrate our model with Clinical Decision Support Systems (CDSS) that operate in real time. This would enhance evidence-based treatment and data reuse by giving clinicians access to clear, understandable suggestions throughout traditional practice. Our method is intended to use knowledge-driven reasoning and adaptive learning to give transparent and customized decision assistance, in contrast to many other CDSS that rely on intractable prediction models or static rules.

In summary, this research provides a foundation that connects modern AI methods in clinical informatics with FAIR data practices. The directions outlined above highlight possible paths for extending the proposed framework toward more robust, interpretable, and scalable medical AI systems that remain aligned with ethical and FAIR principles.

10.4 Clinical Deployment Scope and Ethical Boundaries

The MetaGRO framework, together with the Reinforcement Learning Guided Medicine Optimization (RLGMO) module, should currently be considered a research-oriented clinical decision support system rather than a fully autonomous tool for medicine optimization. For this reason, several limitations and practical considerations must be taken into account before any potential use in real clinical environments.

First, the current model has been evaluated using retrospective datasets. Such datasets are useful for experimental testing and benchmarking, but they do not fully represent real clinical workflows. The framework has not yet been tested using prospective clinical datasets, which would better reflect real-world patient care and treatment conditions.

Second, before any clinical use can be considered, the system must be validated using external datasets. This would require testing under different conditions, including:

- datasets collected from multiple medical institutions;
- patient populations with different demographic characteristics;
- different oncology subtypes and treatment protocols.

Multi-center validation is particularly important in oncology, since treatment approaches and clinical practices may differ across hospitals and geographic regions.

Third, any clinical implementation of such a system must comply with existing regulatory requirements governing medical AI technologies and health data use. In the European Union, AI-based clinical decision support systems must meet the requirements of the Medical Device Regulation (MDR), which regulates software used for medical purposes. In addition, the EU Artificial Intelligence Act (AI Act) classifies many medical AI systems as high-risk applications, requiring strict standards for data quality, transparency, risk management, and human oversight. The processing of clinical data must also comply with the General Data Protection Regulation (GDPR), which ensures the protection of personal and sensitive health data. Furthermore, EU initiatives such as the Data Governance Act promote trustworthy data sharing and support frameworks like the EU register of recognized data altruism organizations to enable responsible data use for research and public interest purposes. (In the United States, comparable systems must obtain approval from the Food and Drug Administration (FDA) before they can be used in clinical practice.)

It is also important to emphasize that the MetaGRO framework is not designed to replace decisions made by oncologists or other healthcare professionals. The system is intended to function as a human-in-the-loop decision support tool. Final treatment decisions must always remain with qualified medical experts who can interpret AI-generated suggestions in the context of the patient's full clinical record, including medical history, diagnostic findings, existing conditions, and other relevant information.

Because cancer treatment involves high clinical risk, additional safeguards would be necessary if the system were to be deployed. These could include continuous monitoring of system behavior, periodic retraining with updated clinical data, and clear documentation of model limitations.

In conclusion, although the MetaGRO framework provides a technically sound approach that combines BERT-based biomedical entity extraction, medical knowledge graph construction, and reinforcement learning for oncology therapy optimization, it remains at the research stage. Further validation, regulatory review, and clinical testing will be required before it can be considered for practical use in healthcare settings.

Appendix A

List of Abbreviations

NER	Named Entity Recognition
BERT	Bidirectional Encoder Representations from Transformers
BC	Breast Cancer
AI	Artificial Intelligence
EHR	Electronic Health Record
ML	Machine Learning
NLP	Natural Language Processing
FL	Federated Learning
E-RNN	Enhanced Recurrent Neural Network
HDRO	Hybrid Dragon-Rider Optimization
DA	Dragonfly Algorithm
RDA	Red Deer Algorithm
ViT	Vision Transformers
BEiT	BERT pre-training of Image Transformers
DWT	Discrete Wavelet Transformation
2D-ICNN	Two-Dimensional Improved Convolutional Neural Network
GLCM	Gray-Level Co-Occurrence Matrix
OSCC	Oral Squamous Cell Carcinoma
RPM	Recurrence Prediction Model
AUC	Area Under the Curve
EM	Expectation–Maximization Algorithm
PICO	Participants, Intervention, Control, and Outcomes
WPSO	Weighted Particle Swarm Optimization
SNF	Similarity Network Fusion

HARP	Hierarchical Representation Learning for Networks
LDA	Latent Dirichlet Allocation
LSTM	Long Short-Term Memory
CAD	Computer-Aided Diagnosis
HMM	Hidden Markov Model
CRF	Conditional Random Field
RLGMO	Reinforcement Learning Graph-based Medicine Optimization
KGPA	Knowledge Graph Pattern Analysis
GNN	Graph Neural Network
GCN	Graph Convolutional Network
R-GCN	Relational Graph Convolutional Network
GAT	Graph Attention Network
RL	Reinforcement Learning
DQL	Deep Q-Learning
ODE	Ordinary Differential Equation
Fastformer	Fast Lightweight Transformer for Sequential Modeling
CNN	Convolutional Neural Network
RNN	Recurrent Neural Network
Node2Vec	Node Embedding Technique for Graphs
Word2Vec	Word Embedding Technique from NLP
PR-AUC	Precision–Recall Area Under the Curve
Jaccard Score	Similarity Measure for Multi-label Classification
DrugBank	Biomedical Database for Drug and Drug-Target Information
MIMIC-III	Medical Information Mart for Intensive Care III
MIMIC-IV	Medical Information Mart for Intensive Care IV
ICD-9	International Classification of Diseases, Ninth Revision
DDI	Drug–Drug Interaction
ADR	Adverse Drug Reaction
KGDNet	Knowledge Graph-Driven Medicine Recommendation Network
SMR	Safe Medicine Recommendation

CompNet	Combined Order-Free Medicine Prediction Network
REST	Reinforced Student–Teacher Curriculum Learning
DGCL	Distance-wise and Graph Contrastive Learning
MERITS	Medication Recommendation on Irregular Time-Series
SIET	Star Interactive Enhanced Transformer
MIFNet	Multimodal Interactive Fusion Network
PHL	Personal Health Library

Appendix B

Algorithms

Algorithm 1: Zero-shot Entity Extraction Pipeline (BioBERT diseases NER)

Input: Seeds $\mathcal{S} = \{42, 56, 101, 202, 303\}$; directories $\mathcal{D} = \{$/breastca, /pdac$\}$; model $m =$`alvaroalon2/biobert_diseases_ner`
Output: CSV files `extracted_entities_with_biobert_seed{s}.csv`

Main:
tokenizer $T \leftarrow$ AUTOTOKENIZER.FROM_PRETRAINED(m)
foreach $s \in \mathcal{S}$ **do**
    SETSEED(s)
    model $M \leftarrow$ AUTOMODELFORTOKENCLASSIFICATION.FROM_PRETRAINED(m)
    PROCESSDIRECTORY($\mathcal{D}, T, M, s,$ `extracted_entities_with_biobert_seed`s`.csv`)
Procedure PROCESSDIRECTORY($\mathcal{D}, T, M, s, out\_csv$):
    results $\mathcal{R} \leftarrow \emptyset$
    foreach *directory* $d \in \mathcal{D}$ **do**
        foreach *file* f *with suffix* `.txt` *in* d **do**
            text $\leftarrow$ READ(f)
            ents $\leftarrow$ EXTRACTENTITIES(text, T, M)
            append {ents, Filename=f, Seed=s} to $\mathcal{R}$
    WRITECSV($\mathcal{R}, out\_csv$)
Function EXTRACTENTITIES(text, T, M): `// returns dict`
    $u \leftarrow$ CLEANTEXT(text)
    $(\mathcal{D}^\star, \mathcal{S}^\star) \leftarrow$ EXTRACTDISEASESSYMPTOMS(u, T, M)
    meds $\leftarrow$ EXTRACTMEDICATIONS(u);
    age $\leftarrow$ EXTRACTAGE(u);
    gender $\leftarrow$ EXTRACTGENDER(u);
    stage $\leftarrow$ EXTRACTCANCERSTAGE(u);
    dose $\leftarrow$ EXTRACTDOSES(u)
    return {Age=age, Gender=gender, *Cancer stage*=stage, Disease=uniq($\mathcal{D}^\star$), Symptoms=uniq($\mathcal{S}^\star$), Medication=meds, Dose=dose}
Function EXTRACTDISEASESSYMPTOMS(u, T, M): `// label 1 = Disease, 2 = Symptom`
    $\mathcal{D}^\star \leftarrow \emptyset, \quad \mathcal{S}^\star \leftarrow \emptyset$
    chunks $\leftarrow$ SPLITCHUNKS(u, max_len $= 512$, overlap $= 50$)
    foreach *chunk* c *in chunks* **do**
        $X \leftarrow T(c,$ truncation $=$ True, max_length $= 512,$ return_tensors $=$ pt$)$
        logits $\leftarrow M(X), \quad \hat{y} \leftarrow \arg\max(\text{logits}, \text{dim} = 2)$
        tokens $\leftarrow T$.convert_ids_to_tokens(X.input_ids)
        $\mathcal{D}^\star \leftarrow \mathcal{D}^\star \cup$ CLEANTOKENS(tokens, $\hat{y}$, target $= 1$)
        $\mathcal{S}^\star \leftarrow \mathcal{S}^\star \cup$ CLEANTOKENS(tokens, $\hat{y}$, target $= 2$)
    return $(\mathcal{D}^\star, \mathcal{S}^\star)$
Function summaries (roles only):
    CLEANTEXT(t) `// Lowercase, collapse spaces, remove non [a-zA-Z0-9\s/.]`
    SPLITCHUNKS(t, max_len, overlap) `// Split into overlapping whitespace-token windows`
    CLEANTOKENS(tokens, labels, target) `// Merge WordPiece ‘‘##’’ fragments; collect contiguous target spans`
    EXTRACTMEDICATIONS(u) `// Regex for ‘‘Name + number + unit’’ (e.g., DrugName 20 mg)`
    EXTRACTAGE(u) `// Regex for age (e.g., 65 y.o., 65 years old)`
    EXTRACTGENDER(u) `// Regex for male/female/man/woman`
    EXTRACTCANCERSTAGE(u) `// Regex for stages (e.g., Stage II, Stage IV)`
    EXTRACTDOSES(u) `// Regex for medication-dose phrases (name + mg)`
    EXTRACTMEDICALHISTORY(u) `// Sentences containing history keywords`

Algorithm 2: Zero-shot Entity Extraction Pipeline (BioClinicalBERT, PubMedBERT, BlueBERT, and RoBERTa)

Input: Seeds $\mathcal{S} = \{42, 56, 101, 202, 303\}$; directories $\mathcal{D} = \{$/breastca, /pdac$\}$; models $m \in \{$
`emilyalsentzer/Bio_ClinicalBERT`,
`microsoft/BiomedNLP-PubMedBERT-base-uncased-abstract-fulltext`,
`bionlp/bluebert_pubmed_mimic_uncased_L-12_H-768_A-12`,
`Jean-Baptiste/roberta-large-ner-english`$\}$

Output: CSV files `extracted_entities_with_model_name_seed{s}.csv`

Main:
foreach *model m in model set* **do**
  tokenizer $T \leftarrow$ AUTOTOKENIZER.FROM_PRETRAINED(m)
  foreach $s \in \mathcal{S}$ **do**
    SETSEED(s)
    model $M \leftarrow$ AUTOMODELFORTOKENCLASSIFICATION.FROM_PRETRAINED(m)
    PROCESSDIRECTORY($\mathcal{D}$, T, M, s, `extracted_entities_with_model_name_seeds.csv`)

Procedure PROCESSDIRECTORY($\mathcal{D}$, T, M, s, out_csv):
  results $\mathcal{R} \leftarrow \emptyset$
  foreach *directory* $d \in \mathcal{D}$ **do**
    foreach *file* f *with suffix* `.txt` *in* d **do**
      text $\leftarrow$ READ(f)
      ents $\leftarrow$ EXTRACTENTITIES(text, T, M)
      append {ents, Filename=f, Seed=s} to $\mathcal{R}$
  WRITECSV($\mathcal{R}$, out_csv)

Function summaries (reused for all models):
  CLEANTEXT(t) `// Normalize text: lowercase, clean non-alphanumeric chars`
  SPLITCHUNKS(t, max_len, overlap) `// Divide text into overlapping token windows`
  CLEANTOKENS(tokens, labels, target) `// Merge ''##'' subwords, group contiguous spans`
  EXTRACTDISEASESSYMPTOMS(u, T, M) `// Model predicts disease and symptom tokens`
  EXTRACTMEDICATIONS(u) `// Regex for DrugName + unit (e.g., ''Tamoxifen 20 mg'')`
  EXTRACTAGE(u), EXTRACTGENDER(u), EXTRACTCANCERSTAGE(u), EXTRACTDOSES(u) `// Entity-specific regex extraction`
  EXTRACTMEDICALHISTORY(u) `// Sentence-level filtering using ''history'' keywords`

Algorithm 3: Fine-tuning BioBERT for Clinical Entity Extraction (Token Classification)

Input: Dataset $\mathcal{D}$ of tokenized clinical sentences with BIO labels;
Pre-trained model $M_0 \leftarrow$ `dmis-lab/biobert-base-cased-v1.1`;
Label set $\mathcal{Y}$ (e.g., {O, B-DISEASE, I-DISEASE, ...});
Hyperparameters: epochs E, batch size B, max length $L_{\max}$, learning rate η, weight decay λ, warmup ratio ρ, random seed set $\mathcal{S}$.
Output: Best fine-tuned model $\hat{M}$, tokenizer $\hat{T}$, and eval metrics (Precision, Recall, F1, Accuracy) averaged over seeds.

Initialize: tokenizer $T \leftarrow$ `AutoTokenizer`(BioBERT vocab);
id2label, label2id $\leftarrow$ mappings from $\mathcal{Y}$;
results $\leftarrow [\,]$.
foreach $s \in \mathcal{S}$ **do**
  `// 1) Reproducibility`
  Set all PRNG seeds to s (Python, NumPy, PyTorch, CUDA)
  `// 2) Preprocess & align labels`
  foreach *example* $(\mathbf{w}, \mathbf{y}) \in \mathcal{D}$ **do**
    (input_ids, attention_mask, word_ids) $\leftarrow T(\mathbf{w}, \textit{is\_split\_into\_words} = \textit{true}, \textit{trunc} = \textit{true}, \textit{max\_length} = L_{\max})$
    `// Assign BIO label to the first subword; mask others with` -100
    $\tilde{\mathbf{y}} \leftarrow$ ALIGNLABELS$(\mathbf{y}$, word_ids$)$
    Append (input_ids, attention_mask, $\tilde{\mathbf{y}}$) to processed dataset
  `// 3) Split train/val`
  $(\mathcal{D}_{\text{train}}, \mathcal{D}_{\text{val}}) \leftarrow$ SPLIT$(\mathcal{D}$, test_size $= 0.15$, random_state $= s)$
  `// 4) Load model & training setup`
  $M \leftarrow$ `AutoModelForTokenClassification`$(M_0, \textit{num\_labels} = |\mathcal{Y}|, \textit{id2label}, \textit{label2id})$
  Optimizer $\leftarrow$ AdamW$(\eta$, weight_decay $= \lambda)$
  Scheduler $\leftarrow$ linear LR with warmup steps $= \rho \times$ total_steps
  Data collator $\leftarrow$ `DataCollatorForTokenClassification`(T)
  `// 5) Train for` E `epochs`
  for $e \leftarrow 1$ **to** E **do**
    foreach *mini-batch* $\mathcal{B} \subset \mathcal{D}_{train}$ *of size* B **do**
      Compute loss (cross-entropy, ignoring labels $= -100$)
      Backpropagate; optimizer step; scheduler step; zero grads
    Evaluate on $\mathcal{D}_{\text{val}}$ using seqeval (entity-level)
    Save checkpoint if validation F1 improves
  `// 6) Select best checkpoint and log metrics`
  (Prec, Rec, F1, Acc) $\leftarrow$ best validation metrics
  Append metrics to results
`// 7) Aggregate across seeds`
$\hat{M} \leftarrow$ checkpoint with highest mean F1; $\hat{T} \leftarrow T$
Report mean $\pm$ std over seeds for (Precision, Recall, F1, Accuracy)
return $\hat{M}, \hat{T}$ and aggregated metrics

Algorithm 4: Entity Extraction after Fine-tuning (Common Inference Pipeline)

Input: Fine-tuned model $\hat{M}$, tokenizer T, label map id2label, text T_x, max length $L_{\max}$, stride s.

Output: Set of entities $\mathcal{E} = \{(label, start, end, value)\}$.

Tokenize T_x without specials to obtain input IDs and offset mapping

Create overlapping windows of size $\leq L_{\max} - 2$ with stride s

foreach *window* **do**

  Add [CLS], [SEP]; run $\hat{M} \rightarrow$ logits; take argmax $\rightarrow$ label IDs; map to BIO tags

  Remove specials; merge subwords by first token; repair invalid BIO transitions

  Convert BIO tags to spans: start on $B-X$, extend through $I-X$, close on O or next B

  Record each $(label, start, end, T_x[start : end])$ span

end

Merge overlapping spans of same label and remove duplicates

Return cleaned entity set $\mathcal{E}$

Algorithm 5: Fine-tuning BioClinicalBERT for Clinical Entity Extraction (Token Classification)

Input: Labeled clinical sentences $\mathcal{D}$ with BIO tags; label set $\mathcal{Y}$; pre-trained model $M_0 =$ `emilyalsentzer/Bio_ClinicalBERT`; epochs E=20; batch size B; learning rate η; weight decay λ; warmup ratio ρ; seed set $\mathcal{S}$ with $|\mathcal{S}|$=5.

Output: Fine-tuned model $\hat{M}$ and metrics (Precision, Recall, F1, Accuracy) averaged over seeds.

Load tokenizer T from M_0 and create id2label, label2id from $\mathcal{Y}$
foreach $s \in \mathcal{S}$ **do**
    Set all random seeds to s
    Tokenize each sentence with T using *is_split_into_words*=true and align labels so only the first subword keeps the BIO tag and others are set to -100
    Split the processed data into train/validation (70/15) with *random_state=s*
    Initialize $M \leftarrow$ `AutoModelForTokenClassification`$(M_0,\ num\_labels = |\mathcal{Y}|,\ \text{id2label},\ \text{label2id})$
    Set optimizer AdamW$(M,\ \eta,\ \text{weight_decay} = \lambda)$ and linear scheduler with warmup ratio ρ
    for $e \leftarrow 1$ **to** E **do**
        foreach *batch* $\mathcal{B}$ *from train set of size* B **do**
            Compute cross-entropy loss on token labels (ignore index -100), backpropagate, optimizer step, scheduler step, zero gradients
        end
        Evaluate on validation set with entity-level metrics (seqeval) and keep a checkpoint if F1 improves
    end
    Record the best checkpoint and its metrics for seed s
end
Select the checkpoint with the highest mean F1 across seeds as $\hat{M}$; report mean $\pm$ std of all metrics over seeds
return $\hat{M}$ and aggregated metrics

Algorithm 6: Fine-tuning (BlueBERT / PubMedBERT / RoBERTa) for Clinical NER

Input: Labeled corpus $\mathcal{D}$ with BIO tags; label set $\mathcal{Y}$;
Model $M_0 \in \{$`bionlp/bluebert_pubmed_mimic_uncased_L-12_H-768_A-12`, `microsoft/BiomedNLP-PubMedBERT-base-uncased-abstract`, `roberta-base`$\}$;
Epochs E=20; batch size $B \in [16, 32]$; learning rate 5×10^{-5};
weight decay 0.01; warmup steps 300; dropout 0.1;
seed set $\mathcal{S} = \{42, 56, 101, 202, 303\}$.
Output: Best fine-tuned model $\hat{M}$ and validation metrics (Precision, Recall, F1).

Tokenize corpus with $T \leftarrow$ `AutoTokenizer`(M_0); align BIO labels (first subword keeps tag, others $\rightarrow -100$)
Split $\mathcal{D}$ into train/val/test with ratios 70:15:15
foreach $s \in \mathcal{S}$ **do**
    SetSeeds(s)
    $M \leftarrow$ `AutoModelForTokenClassification`$(M_0,\ num\_labels = |\mathcal{Y}|)$; apply dropout 0.1
    Optimizer $\leftarrow$ AdamW(M, lr=5×10^{-5}, weight_decay=0.01)
    Scheduler $\leftarrow$ linear warmup (300 warmup steps, total $\approx$ 3000)
    for $e \leftarrow 1$ **to** E **do**
        Train on mini-batches of size B; compute cross-entropy loss (ignore -100)
        Backpropagate, optimizer step, scheduler step, zero gradients
        Evaluate entity-level F1 on validation set; save checkpoint if F1 improves
    end
    Record best metrics for seed s
end
Select checkpoint with highest mean F1 across seeds as $\hat{M}$; report mean $\pm$ std of metrics
return $\hat{M}$ and averaged results

Algorithm 7: Evaluation of Clinical NER Models (Entity-level Metrics)

Input: Ground truth annotations $\mathcal{G}$, predicted entities $\mathcal{P}$, label set $\mathcal{Y}$.
Output: Precision, Recall, F1 (micro and macro).

Normalize each entity in $\mathcal{G}$ and $\mathcal{P}$ as $(label, start, end)$ tuples
$TP \leftarrow$ count of exact label-span matches; $FP \leftarrow$ predicted spans not in gold; $FN \leftarrow$ gold spans not predicted
Compute $Precision = \frac{TP}{TP+FP}$, $Recall = \frac{TP}{TP+FN}$, $F1 = \frac{2PR}{P+R}$
Compute per-label metrics by restricting counts to each $Y \in \mathcal{Y}$; average across labels (macro)
Return full classification report with per-entity and overall scores

Algorithm 8: Construct Knowledge Graph from Structured Cancer Clinical Data

Require: Dataset $\mathcal{D}$ containing patient records (CSV), reference guidelines $\mathcal{G}$
Ensure: Knowledge graph $G = (V, E)$
$G \leftarrow (\varnothing, \varnothing)$
Define entity types
  $\mathcal{T} = \{\texttt{RLGMO}, \texttt{Symptom}, \texttt{Disease}, \texttt{Medication}, \texttt{Dose}, \texttt{Stage}, \texttt{Reference}\}$
foreach *RLGMO record* $r \in \mathcal{D}$ **do**
  Add node $r.\texttt{RLGMO\_ID}$ with type `Patient`
  foreach *entity type* $e \in \{\texttt{Symptom}, \texttt{Disease}, \texttt{Medication}, \texttt{Dose}\}$ **do**
    foreach *value* $v \in r[e]$ **do**
      if $v \neq \varnothing$ **then**
        Add node v with type e
        Add edge $(r.\texttt{RLGMO\_ID}, v)$
  if $r.\texttt{Cancer\_Stage} \neq \varnothing$ **then**
    Add node $r.\texttt{Cancer\_Stage}$ with type `Stage`
    Add edge $(r.\texttt{RLGMO\_ID}, r.\texttt{Cancer\_Stage})$
  if $r.\texttt{Medical\_History} \neq \varnothing$ **then**
    foreach *value* $h \in r.\texttt{Medical\_History}$ **do**
      Add node h with type `Symptom`
      Add edge $(r.\texttt{RLGMO\_ID}, h)$
foreach *guideline* $(\mathcal{T}, t, id) \in \mathcal{G}$ **do**
  Add node $\texttt{REF\_}id$ with type `Reference`
  foreach *term* $x \in \mathcal{T}$ **do**
    Add edge $(\texttt{REF\_}id, x)$
  if $t \neq \varnothing$ **then**
    Add edge $(\texttt{REF\_}id, t)$
return G

Algorithm 9: GCN Training for Patient Embedding

Require: Feature matrix $X \in \mathbb{R}^{N \times d_{\text{in}}}$, edge list E, epochs T, hidden size d_{hidden}, output size d_{out}, optimizer $\mathcal{O}$, learning rate η
Ensure: Trained GCN parameters θ, RLGMO embeddings $Z_p \in \mathbb{R}^{N \times d_{\text{out}}}$
$X \leftarrow$ `Normalize`(X) `// Normalize selected feature columns (e.g., age, stage)`
$A \leftarrow$ `BuildEdgeIndex`(E) `// Build edge-index / adjacency representation`
Initialize GCN parameters θ with $(d_{\text{in}}, d_{\text{hidden}}, d_{\text{out}})$
for $t \leftarrow 1$ **to** T **do**
    $Z \leftarrow$ `GCN`$(X, A; \theta)$ `// Forward pass`
    $\mathcal{L} \leftarrow \text{MSE}(Z, X)$ `// Self-supervised reconstruction (example)`
    $\mathcal{O}$.zero_grad()
    Compute gradients $\nabla_\theta \mathcal{L}$ `// Backpropagation`
    Update parameters $\theta \leftarrow \mathcal{O}.\text{step}(\theta, \nabla_\theta \mathcal{L})$
$Z_p \leftarrow Z$ `// Final node between RLGMO embeddings`
Save Z_p and RLGMO-to-index mapping
return (θ, Z_p)

Algorithm 10: Deep Q-Learning Training Loop for Therapy Recommendation

Require: Replay buffer $\mathcal{B}$, Q-network $Q(s, a; \theta)$, target network $Q(s, a; \theta^-)$
Require: Learning rate α, discount factor γ, batch size N, exploration rate ϵ, update frequency C
Ensure: Trained Q-network parameters θ
Initialize replay buffer $\mathcal{B}$
Randomly initialize Q-network parameters θ and set target network $\theta^- \leftarrow \theta$
for *each episode* $e = 1$ **to** E **do**
    Reset environment and obtain initial state s_0
    for *each step* $t = 1$ **to** T **do**
        With probability ϵ select a random action a_t; otherwise choose $a_t = \arg\max_a Q(s_t, a; \theta)$
        Execute action a_t and observe reward r_t, next state s_{t+1}, and info
        Store transition (s_t, a_t, r_t, s_{t+1}) in $\mathcal{B}$
        Sample random minibatch of N transitions (s, a, r, s') from $\mathcal{B}$
        foreach *sampled transition* (s, a, r, s') **do**
            Compute target $y = r + \gamma \max_{a'} Q(s', a'; \theta^-)$
            Compute loss $L = (y - Q(s, a; \theta))^2$
            Update $\theta \leftarrow \theta - \alpha \nabla_\theta L$
        if $t \bmod C = 0$ **then**
            Update target network $\theta^- \leftarrow \theta$
return Optimized Q-network $Q(s, a; \theta)$

Algorithm 11: Reinforcement Learning Step in `MedicalRecommendationEnv`

Require: Selected action a, current RLGMO_ID p
Ensure: Next state, reward r, done flag, info
Retrieve RLGMO embedding vector v_p from GNN
Map action a to medication m
Extract disease, symptom, and medication sets of RLGMO p
foreach *RLGMO* $q \in \mathcal{P}$, $q \neq p$ **do**
    if *embedding exists and* $(disease/symptom)$ *shared with* p *and* m *in medications of* q **then**
        $sim \leftarrow \texttt{CosineSimilarity}(v_p, v_q)$
        $r \leftarrow \min(1.0,\ 0.6 + 0.1 \times |shared\_terms| + 0.3 \times sim)$
        Mark match type as `similar_RLGMO`
        break
if *no similar RLGMO_ID found* **then**
    if m *in actual medications of* p **then**
        $r \leftarrow 0.8$; match type $\leftarrow$ `exact`
    else if m *is class-equivalent via RxNorm* **then**
        $r \leftarrow 0.6$; match type $\leftarrow$ `class_equivalent`
    else if m *matches reference guideline* **then**
        $r \leftarrow 0.6$; match type $\leftarrow$ `guideline`
    else
        if *diseases or symptoms exist* **then**
            $r \leftarrow 0.3$; match type $\leftarrow$ `partial`
        else
            $r \leftarrow 0.0$; match type $\leftarrow$ `no_info`
return Next state, reward r, `done = true`, info

Algorithm 12: Findability Assessment for a Dataset

Input: Dataset D and its metadata M
Output: Status $\in \{\text{Compliant}, \text{ActionRequired}\}$ and Action List $\mathcal{A}$

Function `AssessFindability`(D, M)**:**
  $\mathcal{A} \leftarrow \emptyset$
  if *Dataset D is* ***not*** *shared* **then**
    if *no alternative access method exists (e.g., local/controlled)* **then**
      add "Provide justification why dataset not available (e.g., regulation)" to $\mathcal{A}$
      return *ActionRequired*, $\mathcal{A}$
    else
      if *D is* ***not*** *fully accessible via the stated method* **then**
        add "Describe information on how to access" to $\mathcal{A}$
        return *ActionRequired*, $\mathcal{A}$
      else
        add "Describe how to access the data" to $\mathcal{A}$
        return *ActionRequired*, $\mathcal{A}$
  if *D* ***does not*** *contain all possible instances* **then**
    add "Provide justification and add the dataset details" to $\mathcal{A}$
    return *ActionRequired*, $\mathcal{A}$
  if *metadata M* ***not*** *provided* **then**
    add "Provide indexed identifier or PID" to $\mathcal{A}$
    return *ActionRequired*, $\mathcal{A}$
  if *raw data* ***does not*** *follow identifier schema* **then**
    add "Provide link for persistent metadata" to $\mathcal{A}$
    return *ActionRequired*, $\mathcal{A}$
  if *persistent metadata datalink/authority link* ***not*** *provided* **then**
    add "Follow data format & type description and provide details" to $\mathcal{A}$
    return *ActionRequired*, $\mathcal{A}$
  if *metadata/semantic validator* ***not*** *followed* **then**
    add "Provide preprocessed metadata" to $\mathcal{A}$
    return *ActionRequired*, $\mathcal{A}$
  if *"raw" data* ***not*** *saved in addition to preprocessed/cleaned/labeled* **then**
    add "Provide preprocessed metadata" to $\mathcal{A}$
    return *ActionRequired*, $\mathcal{A}$
  add "Provide the link or other access point" to $\mathcal{A}$
  return *Compliant*, $\mathcal{A}$

Algorithm 13: Accessibility Assessment for a Dataset

Input: Dataset D, metadata M
Output: Status $\in \{\mathsf{Compliant}, \mathsf{ActionRequired}\}$, action list $\mathcal{A}$

$\mathcal{A} \leftarrow \emptyset$
if *protocol is not open, free, and universally implementable* **then**
    add "Adopt an open, free, universally implementable protocol" to $\mathcal{A}$
if *authorization/authentication not followed where necessary* **then**
    add "Ensure authorization and authentication where necessary" to $\mathcal{A}$
if *metadata not retrievable by identifier via a standardized communications protocol* **then**
    if *no other access methods (e.g., local) exist* **then**
        add "Provide justification for unavailability (e.g., regulation)" to $\mathcal{A}$
        return *ActionRequired*, $\mathcal{A}$
    else
        if *D not fully accessible* **then**
            add "Describe which information is accessible and how to access it" to $\mathcal{A}$
            return *ActionRequired*, $\mathcal{A}$
        else
            add "Describe how to access the data" to $\mathcal{A}$
            return *ActionRequired*, $\mathcal{A}$
if *repository does not follow quality guidelines for stored data and metadata* **then**
    add "Provide metadata aligned with quality criteria" to $\mathcal{A}$
    return *ActionRequired*, $\mathcal{A}$
if *data security and services not provided* **then**
    add "Provide the dataset protected by legal privilege" to $\mathcal{A}$
    return *ActionRequired*, $\mathcal{A}$
if *metadata not accessible when data are no longer available* **then**
    add "Provide raw dataset, tool version, and all configuration parameters" to $\mathcal{A}$
    return *ActionRequired*, $\mathcal{A}$
if *data access is restricted* **then**
    add "Provide data with sharing beyond legal restrictions" to $\mathcal{A}$
else
    add "Provide data with justification for data access restriction" to $\mathcal{A}$
return *Compliant*, $\mathcal{A}$

Algorithm 14: Interoperability Assessment for a Dataset

Input: Dataset D, metadata M
Output: Status $\in \{\mathsf{Compliant}, \mathsf{ActionRequired}\}$, action list $\mathcal{A}$

$\mathcal{A} \leftarrow \emptyset$
if *D not interoperable* **then**
    add "Provide justification on why the dataset is not interoperable" to $\mathcal{A}$
    return *ActionRequired*, $\mathcal{A}$
if *metadata not using a formal, accessible, shared, and broadly applicable language* **then**
    add "Provide reference of the metadata" to $\mathcal{A}$
    return *ActionRequired*, $\mathcal{A}$
if *no proper standard vocabularies, thesaurus, ontologies, or data dictionary provided* **then**
    add "Describe more information about dataset/metadata following interoperability criteria" to $\mathcal{A}$
    return *ActionRequired*, $\mathcal{A}$
if *proper quality of data model not provided* **then**
    add "Provide quality level of access for dataset description" to $\mathcal{A}$
    return *ActionRequired*, $\mathcal{A}$
add "Provide all details of the raw dataset" to $\mathcal{A}$
return *Compliant*, $\mathcal{A}$

Algorithm 15: Reusability Assessment for a Dataset

Input: Dataset D, metadata M
Output: Status $\in \{\mathsf{Compliant}, \mathsf{ActionRequired}\}$, action list $\mathcal{A}$

$\mathcal{A} \leftarrow \emptyset$
if *D cannot be shared* **then**
  if *no alternative access method exists* **then**
    add "Provide justification on why the dataset is not available (e.g., regulation)" to $\mathcal{A}$
    return *ActionRequired*, $\mathcal{A}$
  else
    if *D not fully accessible* **then**
      add "Describe which information are accessible and how to access them" to $\mathcal{A}$
      return *ActionRequired*, $\mathcal{A}$
    else
      add "Describe how to access the data" to $\mathcal{A}$
      return *ActionRequired*, $\mathcal{A}$
if *D distributed under copyright or license terms* **then**
  add "Describe the ToU or license and provide link or reproduce relevant terms" to $\mathcal{A}$
else
  add "Provide dataset with description of potential reuse and open format of software" to $\mathcal{A}$
if *no action for data reuse potential provided* **then**
  add "Provide dataset with provenance for derived data and describe curation protocol" to $\mathcal{A}$
  return *ActionRequired*, $\mathcal{A}$
if *traceability of D not provided* **then**
  add "Provide dataset with provenance and data curation steps" to $\mathcal{A}$
  return *ActionRequired*, $\mathcal{A}$
if *dataset not following enhancement rules* **then**
  add "Provide dataset and describe data enrichment and quality assurance processes" to $\mathcal{A}$
  return *ActionRequired*, $\mathcal{A}$
if *data reusability tools not publicly available* **then**
  add "Provide dataset with methods and tools ensuring long-term integrity and quality standards" to $\mathcal{A}$
  return *ActionRequired*, $\mathcal{A}$
if *data reusability rights not provided* **then**
  add "Provide dataset with sharing arrangements meeting data ethics and protection requirements" to $\mathcal{A}$
  return *ActionRequired*, $\mathcal{A}$
add "Provide dataset with justification of legal reuse restrictions and verify license compliance" to $\mathcal{A}$
return *Compliant*, $\mathcal{A}$

Appendix C

List of Tables

TABLE C.1: Overview of state-of-the-art approaches for clinical text entity extraction.

Entities/Approaches	Cancer-Related Disease	Cancer-Related Symptoms	Cancer-Related Medication	Cancer [Stage/ Classification]	Medical History	Cancer [Type/ Features]	Cancer Detection	Knowledge Extraction	Gene-Related
Our Approach [BioBERT, BioClinicalBERT, PubMedBERT, RoBERTa, BlueBERT]	(all models Fine-tuned with BIO-tagged entities), NLP [POS Tagging, NLTK Stopword Removal, Tokenization, Lemmatization]	(all models Fine-tuned with BIO-tagged entities), Custom Patterns, and NLP [POS Tagging, NLTK Stopword Removal, Tokenization, Lemmatization]	(all models Fine-tuned with BIO-tagged entities), NLP [POS Tagging, NLTK Stopword Removal, Tokenization, Lemmatization]	(all models Fine-tuned with BIO-tagged entities), NLP [POS Tagging, NLTK Stopword Removal, Tokenization, Lemmatization]	×	×	×	×	×
Gholipour et al.[174]	×	NLP	×	×	×	NLP	×	×	×
Nishioka et al.[113]	×	Deep Learning Models [BERT, ELECTRA, T5]	×	×	×	×	×	×	×
Kumari et al.[175]	×	×	×	VGG-16, Xception, Densenet-201	×	VGG-16, Xception, Densenet-201	×	×	×
Solarte-Pabón et al.[176]	BERT, RoBERTa Biomedical	BERT, RoBERTa Biomedical	RoBERTa Biomedical	×	BETO, Multilingual BERT, RoBERTa Biomedical, RoBERTa BNE	BETO Multilingual BERT, RoBERTa Biomedical, RoBERTa BNE	×	×	×
Alhussan et al.[177]	CNN (AlexNet), Al-Biruni Earth Radius (ABER)	×	×	CNN with transfer learning	×	AlexNet	×	×	×
Kumbhare et al.[178]	×	×	×	E-RNN, HDRO, DA, RDA	×	DenseNet	FL	×	×
Catanuto et al.[179]	×	×	×	×	×	×	×	NLP, Word2Vec	×
Chaudhury et al.[115]	×	×	×	ViT, BEiT, (RNN-LSTM)	×	Vision Transformers (ViT) with BEiT	×	×	×
Zhang et al.[180]	×	×	×	×	×	×	×	×	Bhattacharyya Distance Index and Gini Index, PCA
Nguyen et al.[181]	×	×	×	DWT, Radiomics-based feature extraction, Random Forest, SVM, XGBoost	×	DWT, 4- fold cross-validation	×	×	×
Rashmi et al.[182]	×	×	×	×	×	×	×	TF-IDF-CNN	×
Hong et al.[183]	×	×	×	×	×	Statistical, syntactic, and template matching algorithms, SVM	NN	×	×

Entities/Approaches	Cancer-Related Disease	Cancer-Related Symptoms	Cancer-Related Medication	Cancer [Stage/ Classification]	Medical History	Cancer [Type/ Features]	Cancer Detection	Knowledge Extraction	Gene-Related
Ramya et al.[184]	×	×	×	2D-ICNN, GLCM	×	GLCM	OSCC	×	×
Choi, H.S. et al.[119]	×	×	×	×	×	×	×	LLM using GPT	×
Takeshita et al.[185]	×	×	×	×	×	×	×	×	AI, RPM, AUC
Garcia-Barragan et al.[186]	×	×	×	×	×	×	DL-NLP	×	×
Poudel et al.[187]	Logistic Multi-Class Regression Model, SVC	×	×	×	ML	×	×	Text mining and NLP ×	
Voloincu et al.[116]	×	×	×	×	×	×	×	BioBERT, CNN	×
Zeng et al.[188] ×	×	×	×	×	×	LSTM, XG-Boost, and SVM	×	×	
Guvakova et al.[189]	×	×	×	×	×	×	×	×	g3mclass software, EM algorithm.
Mutinda et al. [118]	×	×	×	×	×	×	×	BERT-based NER (PICO)	×
Doma et al. [190]	×	×	×	AI + Optimization	×	PSO, WPSO	×	×	×
Schiappa et al. [191]	×	×	×	×	×	×	×	RUBY Tool	×
Zhou et al. [111]	×	×	×	CancerBERT	×	BlueBERT Fine-tuning	×	×	Cancer BERT
Ren et al. [192]	×	×	BioChemDDI Framework	×	×	×	×	×	×
Watanabe et al. [117]	×	×	×	×	×	×	×	Fine-tuned BERT	×
Sahoo et al. [193]	×	×	×	×	×	×	CNN (VGG16, ResNet50, Inception-v3)	×	×
Ghorpade et al. [194]	×	×	×	×	×	×	×	NLP, LDA, LSTM	×
Achilonu et al. [195]	×	×	×	Rule-based NLP	Regex, Pattern Std.	×	×	×	Rule-based NLP
Pagad et al. [196]	×	×	×	×	×	×	×	ML + NLP, Data Progression	×
Qumsiyeh et al. [197]	Text Mining, NLP, SciSpaCy	×	Text Mining, NLP, SciSpaCy	×	×	×	×	Text Mining, NLP, SciSpaCy	×
Zhou et al. [198]	×	×	×	×	×	ML (HMM, CRF), DL (BiLSTM-CRF)	ML (HMM, CRF), DL (BiLSTM-CRF)	×	×

TABLE C.2: Overall summary of related works on Medical Knowledge Graphs.

Reference#	Proposed Approach	Findings	Limitations
Swathi M. et al. [121]	Developed a drug recommendation system using GraphSAGE for knowledge graph embeddings, optimized by SalpPSO to fine-tune hyperparameters on clinical datasets (MIMIC-III, DrugBank, ICD-9).	The optimized model improved drug-drug interaction prediction and personalized treatment accuracy, aiding clinical decision-making.	The abstract does not address model generalizability across diverse patient populations or real-world clinical settings.
Mythili R. et al. [124]	Introduced Hetrosim GCN-GAT, a GNN-based model combining GCN and attention mechanisms to predict drug-drug and drug-protein interactions using heterogeneous medical graphs.	Achieved superior performance in drug-drug interaction and protein interaction prediction compared to baseline models.	Real-world applications like node and graph classification were not explored; current work is limited to benchmark datasets and link prediction only.

To be continued...

TABLE C.2: related works on Medical Knowledge Graphs (Continued)

Reference#	Proposed Approach	Findings	Limitations
Wang N. et al. [122]	Developed the SIET model combining EMRs and medical knowledge graphs using a Transformer-based architecture.	Outperformed baseline models in recommending drugs with better handling of side effects.	High computational cost and limited real-world scalability remain challenges.
Symeonidis P. et al. [123]	Merged EHR with DDI graphs using three integration stages for explainable drug recommendations.	Significantly improved drug safety and efficacy—suggesting 34× more synergistic drugs and reducing side effects 2350× more than the baseline.	Model complexity and reliance on multiple knowledge graphs may limit scalability in real-world clinical systems.
Symeonidis P. et al. [199]	Integrated EHR data with an adversarial DDI knowledge graph and applied re-ranking with SVD to improve safe drug combination recommendations.	Reduced toxicity of drug combinations with minimal loss in recommendation efficacy.	Slight trade-off in effectiveness to achieve better safety.
Zheng Z. et al. [200]	Developed a DPR framework using GNNs and drug interaction graphs to recommend personalized drug packages.	Achieved strong results on real-world hospital data, outperforming baseline methods.	Complexity and reliance on pre-training may limit general use in clinical settings.
Fouladvand S. et al. [201]	Applied a heterogeneous GNN to EHRs to predict specialist consultation needs for endocrinology and hematology using link prediction.	Outperformed manual checklists and prior systems, achieving higher ROC-AUC and better precision–recall metrics in both specialties.	Integration into real-world clinical workflows and reliance on specialty-specific data may limit broader applicability.
Li X. et al. [202]	Introduced DGCL, a two-stage neural framework combining patient history learning with graph contrastive learning to reduce harmful DDIs.	Outperformed baseline models on MIMIC-III, improving both safety and accuracy in drug recommendations.	The model's complexity and training demands may limit deployment in real-time or resource-constrained clinical settings.
Zhang S. et al. [203]	MERITS uses Neural ODEs and drug interaction graphs to model irregular EMR data for chronic disease medication recommendations.	Outperformed state-of-the-art models on a diabetes dataset.	Depends on clean, annotated data; real-world EMRs may affect performance.
Kim S. et al. [204]	Designed a GNN-based system to recommend personalized rehabilitation exercises using synthetically generated data for knee disease patients.	Outperformed conventional algorithms in top-N recommendation metrics like precision, recall, and nDCG.	Relies on synthetic patient data due to privacy laws, limiting real-world applicability and validation.
Symeonidis P. et al. [205]	Integrated EHR and adversarial DDI graphs for safe drug recommendation and outcome prediction.	Enhanced treatment safety by identifying harmful drug combinations.	Generalizability to diverse patient groups not addressed.
Ammar N. et al. [125]	Developed a decentralized Personal Health Library (PHL) using the Solid platform to support self-care recommendations for diabetic adults.	PHL prototype enables personalized, privacy-aware health recommendations and fosters third-party app development.	Requires further validation through formative evaluation and clinical trials.
Diaz Ochoa J.G. et al. [206]	Built a patient graph using EHR data and applied GNNs to recommend therapies based on patient similarity and diagnosis clustering.	GNNs outperformed baseline models with a 6.48% F1 score gain and revealed meaningful treatment clusters.	Graph construction is complex and sensitive to ICD coding quality; no interventional studies have been conducted yet.

To be continued...

TABLE C.2: related works on Medical Knowledge Graphs (Continued)

Reference#	Proposed Approach	Findings	Limitations
Lai X. et al. [207]	Combined double attention mechanisms and GNNs to recommend safe drug combinations using diagnosis, surgery, medication history, and drug interaction data.	Improved accuracy by 1.02% and reduced drug interactions by 27% compared to baselines.	Does not account for changes in drug use over time with disease progression.
Symeonidis P. et al. [208]	Integrated patient EHRs with both synergistic and adversarial DDI graphs to recommend drug combinations that enhance efficacy and reduce toxicity.	Demonstrated improved safety and effectiveness of treatments using real-world medical datasets.	Implementation details for real-world clinical integration are not addressed.
Wang S. et al. [209]	CompNet uses reinforcement learning with GNNs to predict safe, order-free drug combinations from EHR and DDI data.	Improved Jaccard and F1 scores on MIMIC-III over recent methods.	Depends on the quality of knowledge graphs; may be complex for real-time use.
Sun Z. et al. [210]	Deep learning model for dynamic patient similarity using EHRs and DDI graphs.	Outperformed baselines in diagnosis and medication tasks with lower DDI rates.	Real-time use may be limited by data complexity.
Huo J. et al. [211]	MIFNet combines structured codes and unstructured clinical text from EHRs using a multimodal fusion network.	Outperformed state-of-the-art models on MIMIC-III for medication recommendation.	Text integration may add complexity and require robust preprocessing.
Minh N. et al. [212]	FastRx combines Fastformer for patient history and GCNs for DDI-aware medication recommendation.	Outperformed state-of-the-art models on the MIMIC-III dataset.	Personalized disease progression is modeled, but external validation is not discussed.
Li X. et al. [213]	REST combines a GNN-based student model with a reinforcement learning teacher to balance DDI prediction.	Improved prediction of rare DDI types on benchmark datasets.	Relies on complex training setup and may require fine-tuning for different domains.
Gong F. et al. [214]	SMR embeds EMRs and medical knowledge graphs into a shared space for safe drug recommendation via link prediction.	Effectively recommends medications while accounting for diagnoses and drug interactions.	Dependent on the completeness and quality of external knowledge graphs.
Zhou F. et al. [215]	Used NLP and graph analytics to model ICD-coded medical histories for ADR prediction using Node2Vec and Word2Vec.	Improved recall and AUROC across models, showing strong potential for predicting ADRs.	Generalizability beyond the CBHS dataset is not discussed.
Mishra R. et al. [216]	KGDNet combines longitudinal EHRs, ontologies, and DDI graphs using GNNs, RNNs, and attention for medication recommendation.	Outperformed baselines on MIMIC-IV across all major metrics; validated via ablation and case studies.	Real-time deployment and computational demands not discussed.
M. Liu et al. [217]	A drug ranking framework using PPO-based actor-critic RL with a Deep & Cross Network and GRU for sequential learning.	Demonstrated clinically meaningful ranking performance using cell-line drug screening data and transferability to the TCGA breast cancer cohort.	Lacks integration of symbolic medical knowledge, guideline compliance, and interpretability; relies heavily on black-box modeling and large data.

Bibliography

[1] G. Lorenzini, L. A. Ossa, S. Milford, B. S. Elger, D. M. Shaw, and E. De Clercq, "The "magical theory" of ai in medicine: Thematic narrative analysis," *JMIR AI*, vol. 3, no. 1, e49795, 2024.

[2] N. Queralt-Rosinach et al., "Applying the fair principles to data in a hospital: Challenges and opportunities in a pandemic," *Journal of biomedical semantics*, vol. 13, no. 1, p. 12, 2022.

[3] N. A. of Sciences, Medicine, G. Affairs, B. on Research Data, and C. on Toward an Open Science Enterprise, "Open science by design: Realizing a vision for 21st century research," 2018.

[4] M. Assante et al., "Enacting open science by d4science," *Future Generation Computer Systems*, vol. 101, pp. 555–563, 2019.

[5] C. Xiao, E. Choi, and J. Sun, "Opportunities and challenges in developing deep learning models using electronic health records data: A systematic review," *Journal of the American Medical Informatics Association*, vol. 25, no. 10, pp. 1419–1428, 2018.

[6] H. B. Burke et al., "Electronic health records improve clinical note quality," *Journal of the American Medical Informatics Association*, vol. 22, no. 1, pp. 199–205, 2015.

[7] Y. Luo, P. Szolovits, A. S. Dighe, and J. M. Baron, "Using machine learning to predict laboratory test results," *American journal of clinical pathology*, vol. 145, no. 6, pp. 778–788, 2016.

[8] P. M. Nadkarni, L. Ohno-Machado, and W. W. Chapman, "Natural language processing: An introduction," *Journal of the American Medical Informatics Association*, vol. 18, no. 5, pp. 544–551, 2011.

[9] F. Li, "Named entity recognition," in *Artificial Intelligence and Mobile Services–AIMS 2020: 9th International Conference, Held as Part of the Services Conference Federation, SCF 2020, Honolulu, HI, USA, September 18-20, 2020, Proceedings*, Springer Nature, vol. 12401, 2020, p. 74.

[10] G. Hripcsak, C. Friedman, P. O. Alderson, W. DuMouchel, S. B. Johnson, and P. D. Clayton, "Unlocking clinical data from narrative reports: A study of natural language processing," *Annals of internal medicine*, vol. 122, no. 9, pp. 681–688, 1995.

[11] W. W. Fleuren and W. Alkema, "Application of text mining in the biomedical domain," *Methods*, vol. 74, pp. 97–106, 2015.

[12] A. T. McCray, "An upper-level ontology for the biomedical domain," *Comparative and Functional genomics*, vol. 4, no. 1, pp. 80–84, 2003.

[13] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE transactions on neural networks*, vol. 20, no. 1, pp. 61–80, 2008.

[14] M. D. Wilkinson et al., "The fair guiding principles for scientific data management and stewardship," *Scientific data*, vol. 3, no. 1, pp. 1–9, 2016.

[15] J. S. Ross, R. Lehman, and C. P. Gross, "The importance of clinical trial data sharing: Toward more open science," *Circulation: Cardiovascular Quality and Outcomes*, vol. 5, no. 2, pp. 238–240, 2012.

[16] N. N. K. Krisnawijaya, B. Tekinerdogan, C. Catal, R. van der Tol, and Y. Herdiyeni, "Implementing fair principles in data management systems: A multi-case study in precision farming," *Computers and Electronics in Agriculture*, vol. 230, p. 109 855, 2025.

[17] K. Watson and D. M. Payne, "Ethical practice in sharing and mining medical data," *Journal of Information, Communication and Ethics in Society*, vol. 19, no. 1, pp. 1–19, 2021.

[18] R. Trasarti, V. Grossi, M. Natilli, and B. Rapisarda, "Sobigdata ri: European integrated infrastructure for social mining and big data analytics.," in *SEBD*, 2022, pp. 117–124.

[19] C. L. Parra-Calderón, F. Sanz, and L. D. McIntosh, "The challenge of the effective implementation of fair principles in biomedical research," *Methods of Information in Medicine*, vol. 59, no. 04/05, pp. 117–118, 2020.

[20] S. M. Pincus, I. M. Gladstone, and R. A. Ehrenkranz, "A regularity statistic for medical data analysis," *Journal of clinical monitoring*, vol. 7, no. 4, pp. 335–345, 1991.

[21] M. Wilkinson, M. Dumontier, I. J. Aalbersberg, and et al., "The FAIR guiding principles for scientific data management and stewardship," *Scientific Data*, vol. 3, p. 160 018, 2016. DOI: 10.1038/sdata.2016.18. [Online]. Available: https://doi.org/10.1038/sdata.2016.18.

[22] M. Thompson, "Making fair easy with fair tools: From creolization to convergence," *Data Intelligence*, vol. 2, no. 1-2, pp. 87–95, 2020.

[23] X. Wilcke, P. Bloem, and V. De Boer, "The knowledge graph as the default data model for learning on heterogeneous knowledge," *Data Science*, vol. 1, no. 1-2, pp. 39–57, 2017.

[24] A. G. Barto, "Reinforcement learning," in *Neural systems for control*, Elsevier, 1997, pp. 7–30.

[25] V. Grossi, B. Rapisarda, F. Giannotti, and D. Pedreschi, "Data science at sobigdata: The european research infrastructure for social mining and big data analytics," *International Journal of Data Science and Analytics*, vol. 6, no. 3, pp. 205–216, 2018.

[26] L. Bai, M. D. Mulvenna, Z. Wang, and R. Bond, "Clinical entity extraction: Comparison between metamap, ctakes, clamp and amazon comprehend medical," in *2021 32nd Irish Signals and Systems Conference (ISSC)*, IEEE, 2021, pp. 1–6.

[27] *Proceedings of the Sixth Message Understanding Conference (MUC-6)*, DARPA, 1995.

[28] *Proceedings of the Seventh Message Understanding Conference (MUC-7)*, DARPA, 1998.

[29] A. R. Aronson, "Effective mapping of biomedical text to the UMLS metathesaurus: The metamap program," in *Proceedings of the AMIA Symposium*, American Medical Informatics Association, 2001, pp. 17–21.

[30] O. Bodenreider, "The unified medical language system (UMLS): Integrating biomedical terminology," *Nucleic Acids Research*, vol. 32, no. suppl_1, pp. D267–D270, 2004. DOI: 10.1093/nar/gkh061.

[31] J.-D. Kim, T. Ohta, Y. Tateisi, and J. Tsujii, "GENIA corpus: A semantically annotated corpus for bio-textmining," *Bioinformatics*, vol. 19, no. suppl_1, pp. i180–i182, 2003. DOI: 10.1093/bioinformatics/btg1023.

[32] O. Uzuner, I. Solti, F. Xia, and E. Cadag, "2010 i2b2/VA challenge on concepts, assertions, and relations in clinical text," *Journal of the American Medical Informatics Association*, vol. 18, no. 5, pp. 552–556, 2011. DOI: 10.1136/amiajnl-2011-000203.

[33] J. Lafferty, A. McCallum, and F. C. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," 2001.

[34] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[35] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[36] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[37] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2013, pp. 3111–3119.

[38] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[39] A. Vaswani et al., "Attention is all you need," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2017, pp. 5998–6008.

[40] A. E. Johnson et al., "Mimic-iii, a freely accessible critical care database," *Scientific Data*, vol. 3, no. 1, p. 160 035, 2016.

[41] J. Lee et al., "Biobert: A pre-trained biomedical language representation model for biomedical text mining," *Bioinformatics*, vol. 36, no. 4, pp. 1234–1240, 2020.

[42] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, Minneapolis, Minnesota: Association for Computational Linguistics, 2019, pp. 4171–4186. DOI: 10.18653/v1/N19-1423.

[43] E. Alsentzer et al., "Publicly available clinical bert embeddings," *Proceedings of the 2nd Clinical Natural Language Processing Workshop*, pp. 72–78, 2019.

[44] Y. Peng, S. Yan, and Z. Lu, "Transfer learning in biomedical natural language processing: An evaluation of bert and elmo on ten benchmarking datasets," *arXiv preprint arXiv:1906.05474*, 2019.

[45] Y. Gu et al., "Domain-specific language model pretraining for biomedical natural language processing," *ACM Transactions on Computing for Healthcare (HEALTH)*, vol. 3, no. 1, pp. 1–23, 2021.

[46] J. Ni et al., "Sentence-t5: Scalable sentence encoders from pre-trained text-to-text models," in *Findings of the association for computational linguistics: ACL 2022*, 2022, pp. 1864–1874.

[47] M. Lewis et al., "Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," in *Proceedings of the 58th annual meeting of the association for computational linguistics*, 2020, pp. 7871–7880.

[48] K. L. Tan, C. P. Lee, K. S. M. Anbananthen, and K. M. Lim, "Roberta-lstm: A hybrid model for sentiment analysis with transformer and recurrent neural network," *IEEE Access*, vol. 10, pp. 21 517–21 525, 2022.

[49] A. Hogan et al., "Knowledge graphs," *ACM Computing Surveys (CSUR)*, vol. 54, no. 4, pp. 1–37, 2021.

[50] S. Ji, S. Pan, E. Cambria, P. Marttinen, and P. S. Yu, "A survey on knowledge graphs: Representation, acquisition, and applications," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 2, pp. 494–514, 2021.

[51] D. N. Nicholson and C. S. Greene, "Knowledge graphs and their applications in biomedical informatics," *Bioinformatics*, vol. 36, no. 24, pp. 5388–5398, 2020.

[52] H. Wang et al., "Knowledge graph convolutional networks for recommender systems," in *The World Wide Web Conference*, 2019, pp. 3307–3313.

[53] X. Wang, T. Gao, Z. Zhu, Z. Liu, J. Li, and M. Sun, "Kepler: A unified model for knowledge embedding and pre-trained language representation," in *Proceedings of the 2020 Transactions of the Association for Computational Linguistics*, 2020, pp. 205–220.

[54] G. Zhou et al., "Sagerec: Graph-based recommender systems via graph convolutional networks," *Knowledge-Based Systems*, vol. 205, p. 106 277, 2020.

[55] Y. Lin, H. Tang, X. Han, Y. Yao, Z. Liu, and M. Sun, "K-bert: Enabling language representation with knowledge graph," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, 2020, pp. 2901–2908.

[56] M. Yasunaga, H. Ren, A. Bosselut, P. Liang, and J. Leskovec, "Qa-gnn: Reasoning with language models and knowledge graphs for question answering," in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2021, pp. 535–546.

[57] L. Yan, T. Zhao, Z. Zhang, Y. Li, and Z. Jiang, "Universal representations of knowledge graphs for reasoning and prediction," *Neural Computing and Applications*, vol. 33, no. 24, pp. 16 877–16 890, 2021.

[58] M. Kan, X. Wang, H. Liu, and W. Zhang, "Multi-hop reasoning with knowledge graphs for question answering," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2021, pp. 3204–3216.

[59] A. Santos, R. Ribeiro, and J. L. Oliveira, "Healthcare knowledge graphs: A novel framework for biomedical data integration and reasoning," *Journal of Biomedical Informatics*, vol. 127, p. 104 011, 2022.

[60] T. Chandak, A. Das, S. Ghosh, and S. Banerjee, "Integrating knowledge graphs and deep learning for interpretable clinical decision support," *Artificial Intelligence in Medicine*, vol. 139, p. 102 509, 2023.

[61] F. Wang, L. P. Casalino, and D. Khullar, "A survey on deep learning for clinical decision support systems," *Computers in Biology and Medicine*, vol. 111, pp. 103–115, 2019.

[62] Z. Cheng, Y. Zhang, Y. Li, and F. Wang, "A survey on deep reinforcement learning for treatment recommendation in healthcare," *Artificial Intelligence in Medicine*, vol. 124, p. 102 228, 2022.

[63] L. Yao, C. Mao, and Y. Luo, "Reinforcement learning for treatment recommendation: Fundamentals, applications, and challenges," *Artificial Intelligence in Medicine*, vol. 104, p. 101 822, 2020.

[64] J. Shang, T. Ma, C. Xiao, and J. Sun, "Gamenet: Graph augmented memory networks for recommending medication combinations," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 1126–1133.

[65] Z. Zhao, T. Ma, C. Xiao, and J. Sun, "Graph-based medication recommendation with heterogeneous graph neural networks," *Briefings in Bioinformatics*, vol. 23, no. 1, bbab464, 2022.

[66] E. S. Berner and S. La Lande, "Clinical decision support systems: Theory and practice," *Springer Science & Business Media*, 2007.

[67] M. P. Sendak et al., "The human body is not a black box: Reinforcing the importance of clinical context in ai-based decision support," *NPJ Digital Medicine*, vol. 3, no. 1, p. 87, 2020.

[68] E. H. Shortliffe and J. J. Cimino, *Computer applications in health care and biomedicine*. Springer Science & Business Media, 2012.

[69] M. Komorowski, L. A. Celi, O. Badawi, A. C. Gordon, and A. A. Faisal, "The artificial intelligence clinician learns optimal treatment strategies for sepsis in intensive care," *Nature Medicine*, vol. 24, no. 11, pp. 1716–1720, 2018.

[70] F. Wang, Y. Zhang, and Y. Zhou, "Supervised reinforcement learning with recurrent neural network for dynamic treatment recommendation," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018, pp. 2447–2456.

[71] Q. Song, J. Shi, C. Xiao, and J. Sun, "Safedrug: Dual molecular graph encoders for safe drug recommendations," in *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, 2021, pp. 1730–1739.

[72] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[73] A. G. Barto, "Reinforcement learning: An introduction. by richard s. sutton," *SIAM Review*, vol. 6, no. 2, p. 423, 2021.

[74] V. Mnih et al., "Playing atari with deep reinforcement learning," *arXiv preprint arXiv:1312.5602*, 2013.

[75] A. Esteva et al., "A guide to deep learning in healthcare," *Nature medicine*, vol. 25, no. 1, pp. 24–29, 2019.

[76] F. Jiang et al., "Artificial intelligence in healthcare: Past, present and future," *Stroke and vascular neurology*, vol. 2, no. 4, pp. 230–243, 2017.

[77] A. Rajkomar, J. Dean, and I. Kohane, "Machine learning in medicine," *New England Journal of Medicine*, vol. 380, no. 14, pp. 1347–1358, 2019.

[78] E. J. Topol, "High-performance medicine: The convergence of human and artificial intelligence," *Nature medicine*, vol. 25, no. 1, pp. 44–56, 2019.

[79] B. Shickel, P. J. Tighe, A. Bihorac, and P. Rashidi, "Deep ehr: A survey of recent advances in deep learning techniques for electronic health record (ehr) analysis," *IEEE journal of biomedical and health informatics*, vol. 22, no. 5, pp. 1589–1604, 2018.

[80] R. Miotto, F. Wang, S. Wang, X. Jiang, and J. T. Dudley, "Deep learning for healthcare: Review, opportunities and challenges," *Briefings in bioinformatics*, vol. 19, no. 6, pp. 1236–1246, 2017.

[81] B. Xu et al., "Applications of artificial intelligence in combating covid-19: A review," *Frontiers in public health*, vol. 8, p. 345, 2020.

[82] Y. Wang et al., "Clinical information extraction applications: A literature review," *Journal of biomedical informatics*, vol. 77, pp. 34–49, 2018.

[83] M. Habibi, L. Weber, M. Neves, D. L. Wiegandt, and U. Leser, "Deep learning with word embeddings improves biomedical named entity recognition," *Bioinformatics*, vol. 33, no. 14, pp. i37–i48, 2017.

[84] F. Liu, J. Chen, A. Jagannatha, and H. Yu, "An overview of named entity recognition in biomedical texts," *Journal of Biomedical Informatics*, vol. 119, p. 103 792, 2021.

[85] X. Wu et al., "Disease named entity recognition in clinical notes using deep learning models," *BMC Medical Informatics and Decision Making*, vol. 22, no. 1, pp. 1–12, 2022.

[86] E. Alsentzer et al., "Publicly available clinical bert embeddings," *arXiv preprint arXiv:1904.03323*, 2019, Preprint.

[87] A. Basaad, S. Basurra, E. Vakaj, A. K. Eldaly, and M. M. Abdelsamea, "A bert-gnn approach for metastatic breast cancer prediction using histopathology reports," *Diagnostics*, vol. 14, no. 13, p. 1365, 2024.

[88] X. Yu, W. Hu, S. Lu, X. Sun, and Z. Yuan, "Biobert based named entity recognition in electronic medical record," in *2019 10th international conference on information technology in medicine and education (ITME)*, IEEE, 2019, pp. 49–52.

[89] K. Huang, J. Altosaar, and R. Ranganath, "Clinicalbert: Modeling clinical notes and predicting hospital readmission," *arXiv preprint arXiv:1904.05342*, 2019.

[90] I. Beltagy, K. Lo, and A. Cohan, "Scibert: A pretrained language model for scientific text," *arXiv preprint arXiv:1903.10676*, 2019.

[91] A. Kumar and S. S. Rathore, "A deep learning based approach to automate clinical coding of electronic health records," in *International Conference on Big Data Analytics*, Springer, 2022, pp. 104–116.

[92] J. Mozafari, A. Fatemi, and P. Moradi, "A method for answer selection using distilbert and important words," in *2020 6th International Conference on Web Research (ICWR)*, IEEE, 2020, pp. 72–76.

[93] Z Lan, "Albert: A lite bert for self-supervised learning of language representations," *arXiv preprint arXiv:1909.11942*, 2019.

[94] Y. Guo, Y. Ge, and A. Sarker, "Detection of medication mentions and medication change events in clinical notes using transformer-based models.," in *MedInfo*, 2023, pp. 685–689.

[95] H. Y. Lin and T.-S. Moh, "Sentiment analysis on covid tweets using covid-twitter-bert with auxiliary sentence approach," in *Proceedings of the 2021 ACM Southeast Conference*, 2021, pp. 234–238.

[96] D. Mamakas, P. Tsotsi, I. Androutsopoulos, and I. Chalkidis, "Processing long legal documents with pre-trained transformers: Modding legalbert and longformer," *arXiv preprint arXiv:2211.00974*, 2022.

[97] L. Martin et al., "Camembert: A tasty french language model," *arXiv preprint arXiv:1911.03894*, 2019.

[98] A. Pathak, S. Kumar, P. P. Roy, and B.-G. Kim, "Aspect-based sentiment analysis in hindi language by ensembling pre-trained mbert models," *Electronics*, vol. 10, no. 21, p. 2641, 2021.

[99] S. Kula and M. Gregor, "Multilingual models for check-worthy social media posts detection," *arXiv preprint arXiv:2408.06737*, 2024.

[100] S. Wang, J. Bian, X. Huang, H. Zhou, and S. Zhu, "Publabeler: Enhancing automatic classification of publications in uniprotkb using protein textual description and pubmedbert," *IEEE Journal of Biomedical and Health Informatics*, pp. 1–9, 2024. DOI: 10.1109/JBHI.2024.3520579.

[101] O. Uzuner, A. Bodnari, S. Shen, T. Forbush, J. Pestian, and B. R. South, "Evaluating the state of the art in coreference resolution for electronic medical records," *Journal of the American Medical Informatics Association*, vol. 19, no. 5, pp. 786–791, 2012.

[102] J. Li et al., "Biocreative v cdr task corpus: A resource for chemical disease relation extraction," *Database*, vol. 2016, 2016.

[103] C.-H. Wei et al., "Assessing the state of the art in biomedical relation extraction: Overview of the biocreative v chemical-disease relation (cdr) task," *Database*, vol. 2016, baw032, 2016.

[104] N. Elhadad, B. R. South, D. M. Martinez, and G. K. Savova, "Share/clef ehealth evaluation lab 2014, task 2: Disorder attributes," in *Working Notes of CLEF*, 2014.

[105] Q. Zhang, P. Wang, F. Liu, B. Tang, and H. Xu, "Overview of bionne 2024: Biomedical nested named entity recognition shared task," in *Proceedings of the BioNLP Workshop*, 2024.

[106] W. Hersh, E. M. Voorhees, K. Roberts, and D. Demner-Fushman, "Overview of the trec 2017 precision medicine track," in *TREC*, 2017.

[107] D. Mowery et al., "Task 2: Share/clef ehealth evaluation lab 2013," in *CLEF (Working Notes)*, Citeseer, 2013.

[108] D. E. Losada, F. Crestani, and J. Parapar, "Overview of erisk 2021: Early risk prediction on the internet," in *CLEF (Working Notes)*, CEUR Workshop Proceedings, 2021.

[109] R. I. Doğan, R. Leaman, and Z. Lu, "Ncbi disease corpus: A resource for disease name recognition and concept normalization," *Journal of Biomedical Informatics*, vol. 47, pp. 1–10, 2014.

[110] Y. Labrak et al., "Drbenchmark: A large language understanding evaluation benchmark for french biomedical domain," *arXiv preprint arXiv:2402.13432*, 2024.

[111] S. Zhou, N. Wang, L. Wang, H. Liu, and R. Zhang, "Cancerbert: A cancer domain-specific language model for extracting breast cancer phenotypes from electronic health records," *Journal of the American Medical Informatics Association*, vol. 29, no. 7, pp. 1208–1216, 2022.

[112] O. Solarte-Pabón et al., "Transformers for extracting breast cancer information from spanish clinical narratives," *Artificial Intelligence in Medicine*, vol. 143, p. 102 625, 2023.

[113] S. Nishioka et al., "Adverse event signal extraction from cancer patients' narratives focusing on impact on their daily-life activities," *Scientific reports*, vol. 13, no. 1, p. 15 516, 2023.

[114] J. Hu, R. Bao, Y. Lin, H. Zhang, and Y. Xiang, "Accurate medical named entity recognition through specialized nlp models," *arXiv preprint arXiv:2412.08255*, 2024.

[115] S. Chaudhury and K. Sau, *Retracted: A bert encoding with recurrent neural network and long-short term memory for breast cancer image classification*, 2023.

[116] M. Volosincu, C. Lupu, D. Trandabat, and D. Gifu, "Fii smart at semeval 2023 task7: Multi-evidence natural language inference for clinical trial data," in *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, 2023, pp. 212–220.

[117] T. Watanabe, S. Yada, E. Aramaki, H. Yajima, H. Kizaki, and S. Hori, "Extracting multiple worries from breast cancer patient blogs using multilabel classification with the natural language processing model bidirectional encoder representations from transformers: Infodemiology study of blogs," *JMIR cancer*, vol. 8, no. 2, e37840, 2022.

[118] F. W. Mutinda, K. Liew, S. Yada, S. Wakamiya, and E. Aramaki, "Automatic data extraction to support meta-analysis statistical analysis: A case study on breast cancer," *BMC Medical Informatics and Decision Making*, vol. 22, no. 1, p. 158, 2022.

[119] H. S. Choi, J. Y. Song, K. H. Shin, J. H. Chang, and B.-S. Jang, "Developing prompts from large language model for extracting clinical information from pathology and ultrasound reports in breast cancer," *Radiation Oncology Journal*, vol. 41, no. 3, p. 209, 2023.

[120] A. Hoerbst and E. Ammenwerth, "Electronic health records," *Methods of Information in Medicine*, vol. 49, no. 4, pp. 320–336, 2010.

[121] G. L. SM, S B, and S Hemalatha, "Data-driven drug treatment: Enhancing clinical decision-making with salppso-optimized graphsage," *Computer Methods in Biomechanics and Biomedical Engineering*, 2024, Epub ahead of print. DOI: 10.1080/10255842.2024.2399012.

[122] N. Wang, X. Cai, L. Yang, and X. Mei, "Safe medicine recommendation via star interactive enhanced-based transformer model," *Computer Biology and Medicine*, vol. 141, p. 105 159, 2022, Epub 2021 Dec 24. DOI: 10.1016/j.compbiomed.2021.105159.

[123] P. Symeonidis, S. Chairistanidis, and M. Zanker, "Safe, effective and explainable drug recommendation based on medical data integration," *User Modeling and User-Adapted Interaction*, vol. 32, pp. 999–1018, 2022. DOI: 10.1007/s11257-022-09342-x.

[124] R. Mythili and N. Parthiban, "Link prediction of biomedical data using hetrosim gcn-gat," in *2023 International Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems (ICSES)*, IEEE, 2023.

[125] N. Ammar et al., "Using a personal health library–enabled mhealth recommender system for self-management of diabetes among underserved populations: Use case for knowledge graphs and linked data," *JMIR Formative Research*, vol. 5, no. 3, e24738, 2021.

[126] M. Sushil, V. E. Kennedy, D. Mandair, B. Y. Miao, T. Zack, and A. J. Butte, "Coral: Expert-curated oncology reports to advance language model inference," *NEJM AI*, vol. 1, no. 4, AIdbp2300110, 2024.

[127] F. Yang et al., "Assessing inter-annotator agreement for medical image segmentation," *IEEE Access*, vol. 11, pp. 21 300–21 312, 2023.

[128] L. Ramshaw and M. Marcus, "Text chunking using transformation-based learning," in *Third workshop on very large corpora*, 1995.

[129] E. F. Tjong Kim Sang and F. De Meulder, "Introduction to the conll-2003 shared task: Language-independent named entity recognition," in *Proceedings of CoNLL-2003*, Association for Computational Linguistics, 2003, pp. 142–147.

[130] D. Jurafsky and J. H. Martin, "Speech and language processing," *Prentice Hall Series in Artificial Intelligence*, 2019, 3rd Edition (Draft), Chapter 4 and Chapter 5. [Online]. Available: `https://web.stanford.edu/~jurafsky/slp3/`.

[131] X. Wu, J. Duan, Y. Pan, and M. Li, "Medical knowledge graph: Data sources, construction, reasoning, and applications," *Big Data Mining and Analytics*, vol. 6, no. 2, pp. 201–217, 2023.

[132] M. Sushil et al., "Coral: Expert-curated oncology reports to advance language model inference," *NEJM AI*, vol. 1, no. 4, AIdbp2300110, 2024.

[133] N. Singh Punn, S. K. Sonbhadra, S. Agarwal, et al., "Recommending best course of treatment based on similarities of prognostic markers," *arXiv e-prints*, arXiv–2107, 2021.

[134] A. Harnoune et al., "Bert based clinical knowledge extraction for biomedical knowledge graph construction and analysis," *Computer Methods and Programs in Biomedicine Update*, vol. 1, p. 100 042, 2021. DOI: `10.1016/j.cmpbup.2021.100042`.

[135] P. Chandak, K. Huang, and M. Zitnik, "Building a knowledge graph to enable precision medicine," *Scientific Data*, vol. 10, no. 1, p. 67, 2023.

[136] K. Meding and T. Hagendorff, "Fairness hacking: The malicious practice of shrouding unfairness in algorithms," *Philosophy & Technology*, vol. 37, no. 1, p. 4, 2024.

[137] A. R. D. Commons, *Fair self assessment tool*, 2018. [Online]. Available: `https://ardc.edu.au/resources/working-with-data/fair-data/fair-self-assessment-tool/`.

[138] A. R. D. Commons, *Fair self assessment tool – documentation*, 2020. [Online]. Available: `https://ardc.edu.au/`.

[139] CSIRO, *5-star data rating tool*, 2021. [Online]. Available: `https://doi.org/10.5281/zenodo.5562714`.

[140] L. Pisani and R. van Horik, "Fairdat: Assessing fair maturity through practical questionnaires," *DANS*, 2021. [Online]. Available: `https://doi.org/10.5281/zenodo.4750673`.

[141] L. Michaelis and et al., "Experiences from compiling a FAIR survey in the german network university medicine," in *SWAT4HCLS*, 2023, pp. –.

[142] A. Trifan and J. L. Oliveira, "Towards a more reproducible biomedical research environment: Endorsement and adoption of the FAIR principles," in *International Joint Conference on Biomedical Engineering Systems and Technologies*, Cham: Springer, 2019, pp. –.

[143] A. Trifan and J. L. Oliveira, "FAIRness in biomedical data discovery," in *HEALTHINF*, 2019, pp. –.

[144] C. Bahlo, "Advancing FAIR agricultural data: The agrefed FAIR assessment tool," *Data Science Journal*, vol. 23, no. 1, 2024.

[145] R. Gao, Y. Ge, and C. Shah, "FAIR: Fairness-aware information retrieval evaluation," *Journal of the Association for Information Science and Technology*, vol. 73, no. 10, pp. 1461–1473, 2022.

[146] S. Jones and et al., "Data management planning: How requirements and solutions are beginning to converge," *Data Intelligence*, vol. 2, no. 1-2, pp. 208–219, 2020.

[147] L. Candela, D. Mangione, and G. Pavone, "The FAIR assessment conundrum: Reflections on tools and metrics," *Data Science Journal*, vol. 23, no. 1, 2024.

[148] T. Weber and D. Kranzlmüller, "How FAIR can you get? image retrieval as a use case to calculate FAIR metrics," in *2018 IEEE 14th International Conference on e-Science (e-Science)*, IEEE, 2018, pp. –.

[149] D. J. B. Clarke et al., "FAIRshake: Toolkit to evaluate the FAIRness of research digital resources," *Cell Systems*, vol. 9, no. 5, pp. 417–421, 2019. DOI: 10.1016/j.cels.2019.09.011.

[150] A. Gaignard and et al., "FAIR-checker: Supporting digital resource findability and reuse with knowledge graphs and semantic web standards," *Journal of Biomedical Semantics*, vol. 14, no. 1, p. 7, 2023.

[151] G. Hiranandani and et al., "Quadratic metric elicitation for fairness and beyond," in *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, PMLR, 2022.

[152] A. Craig and et al., "Dream principles and FAIR metrics from the portal-doors project for the semantic web," in *2019 11th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*, IEEE, 2019, pp. –.

[153] M. D. Wilkinson and et al., "Community-driven governance of FAIRness assessment: An open issue, an open discussion," *Open Research Europe*, vol. 2, 2022.

[154] H. Müller and et al., "BIBBOX, a FAIR toolbox and app store for life science research," *New Biotechnology*, vol. 77, pp. 12–19, 2023.

[155] A. Lien-Talks, "How FAIR is bioarchaeological data: With a particular emphasis on making archaeological science data reusable," *Journal of Computer Applications in Archaeology*, vol. 7, no. 1, 2024.

[156] M. P. P. Feijoó and et al., "Evaluating FAIRness of genomic databases," in *International Conference on Conceptual Modeling*, Cham: Springer, 2020.

[157] M. Thompson and et al., "Making FAIR easy with FAIR tools: From creolization to convergence," *Data Intelligence*, vol. 2, no. 1-2, pp. 87–95, 2020.

[158] N. A. Krans, A. Ammar, P. Nymark, E. L. Willighagen, M. I. Bakker, and J. T. K. Quik, "FAIR assessment tools: Evaluating use and performance," *NanoImpact*, vol. 27, p. 100 402, 2022.

[159] C. Sun, V. Emonet, and M. Dumontier, "A comprehensive comparison of automated FAIRness evaluation tools," in *13th International Conference on Semantic Web Applications and Tools for Health Care and Life Sciences (SWAT4HCLS)*, 2022, pp. 44–53.

[160] K. Peters-von Gehlen, H. Höck, A. Fast, D. Heydebreck, A. Lammert, and H. Thiemann, "Recommendations for discipline-specific FAIRness evaluation derived from applying an ensemble of evaluation tools," *Data Science Journal*, vol. 21, pp. 7–7, 2022.

[161] P. Morville, *Ambient findability: What we find changes who we become*. " O'Reilly Media, Inc.", 2005.

[162] V. Grossi, B. Rapisarda, M. Natilli, and R. Trasarti, "Sobigdata ri: European integrated infrastructure for social mining and big data analytics," in *Proceedings of the 1st International Workshop on Ethics and Legal Issues in Big Data Analytics (ELIBDA)*, ser. CEUR Workshop Proceedings, vol. 3194, 2021, pp. 14–24. [Online]. Available: `https://ceur-ws.org/Vol-3194/paper14.pdf`.

[163] A. Unknown, "Fair principles and zenodo as a tool for accessible data," *Zenodo Repository*, 2020, Accessed July 2025. [Online]. Available: `https://zenodo.org/records/8211123/files/FAIR-principles-Zenodo.pdf`.

[164] N. C. for Biotechnology Information (NCBI), *Sequence read archive (sra) – data storage model and cloud delivery service*, `https://www.ncbi.nlm.nih.gov/sra/docs/sra-data-storage-model/`, Accessed July 2025, 2024.

[165] A. Bianchi, G. d'Aloisio, F. Marzi, and A. Di Marco, "A decision tree to shepherd scientists through data retrievability," *arXiv preprint arXiv:2304.05767*, 2023.

[166] A. Hasnain and D. Rebholz-Schuhmann, "Assessing fair data principles against the 5-star open data principles," in *European Semantic Web Conference*, Springer, 2018, pp. 469–477.

[167] S. Majumder, J. Chakraborty, G. R. Bai, K. T. Stolee, and T. Menzies, "Fair enough: Searching for sufficient measures of fairness," *ACM Transactions on Software Engineering and Methodology*, vol. 32, no. 6, pp. 1–22, 2023.

[168] N. Kanopoulos, N. Vasanthavada, and R. L. Baker, "Design of an image edge detection filter using the sobel operator," *IEEE Journal of solid-state circuits*, vol. 23, no. 2, pp. 358–367, 1988.

[169] K. G. Derpanis, "The harris corner detector," *York University*, vol. 2, pp. 1–2, 2004.

[170] C. S. Kenney, M. Zuliani, and B. Manjunath, "An axiomatic approach to corner detection," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, IEEE, vol. 1, 2005, pp. 191–197.

[171] C. Galamhos, J. Matas, and J. Kittler, "Progressive probabilistic hough transform for line detection," in *Proceedings. 1999 IEEE computer society conference on computer vision and pattern recognition (Cat. No PR00149)*, IEEE, vol. 1, 1999, pp. 554–560.

[172] I. Culjak, D. Abram, T. Pribanic, H. Dzapo, and M. Cifrek, "A brief introduction to opencv," in *2012 proceedings of the 35th international convention MIPRO*, IEEE, 2012, pp. 1725–1730.

[173] F. Pedregosa et al., "Scikit-learn: Machine learning in python," *the Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[174] M. Gholipour, R. Khajouei, P. Amiri, S. Hajesmaeel Gohari, and L. Ahmadian, "Extracting cancer concepts from clinical notes using natural language processing: A systematic review," *BMC bioinformatics*, vol. 24, no. 1, p. 405, 2023.

[175] V. Kumari and R. Ghosh, "A magnification-independent method for breast cancer classification using transfer learning," *Healthcare Analytics*, vol. 3, p. 100 207, 2023.

[176] O. Solarte-Pabón, M. Torrente, A. Garcia-Barragán, M. Provencio, E. Menasalvas, and V. Robles, "Deep learning to extract breast cancer diagnosis concepts," in *2022 IEEE 35th international symposium on computer-based medical systems (CBMS)*, IEEE, 2022, pp. 13–18.

[177] A. A. Alhussan et al., "Classification of breast cancer using transfer learning and advanced al-biruni earth radius optimization," *Biomimetics*, vol. 8, no. 3, p. 270, 2023.

[178] S. Kumbhare, A. B. Kathole, and S. Shinde, "Federated learning aided breast cancer detection with intelligent heuristic-based deep learning framework," *Biomedical Signal Processing and Control*, vol. 86, p. 105 080, 2023.

[179] G. Catanuto et al., "Natural language processing to extract meaningful information from a corpus of written knowledge in breast cancer: Transforming books into data," *Breast Care*, vol. 18, no. 3, pp. 209–212, 2023.

[180] C. Zhang and X. Cao, "Biological gene extraction path based on knowledge graph and natural language processing," *Frontiers in Genetics*, vol. 13, p. 1 086 379, 2023.

[181] T. H. Nguyen, M. T. A. Vo, T. H. H. Nguyen, and N. T. Le, "Improving breast mass classification performance of radiomics-based model by image enhancement with discrete wavelet transformation," in *2023 1st International Conference on Health Science and Technology (ICHST)*, IEEE, 2023, pp. 1–6.

[182] C Rashmi, D. Srinivas, S Prabu, P. S. Kumar, and A. Siddiqua, "Natural language processing in healthcare: Intelligent system for extracting insights from clinical text data," in *2023 International Conference on Evolutionary Algorithms and Soft Computing Techniques (EASCT)*, IEEE, 2023, pp. 1–5.

[183] S.-P. Hong and D. Bhattacharyya, "A study on the improvement of security image analysis capability using artificial intelligence," *Tehnički glasnik*, vol. 17, no. 4, pp. 598–604, 2023.

[184] S. Ramya, R. Minu, and K. Magesh, "An automated approach for identification of oral squamous cell carcinoma based on 2d-icnn," in *2023 Second International Conference on Augmented Intelligence and Sustainable Systems (ICAISS)*, IEEE, 2023, pp. 155–161.

[185] T. Takeshita, H. Iwase, R. Wu, D. Ziazadeh, L. Yan, and K. Takabe, "Development of a machine learning-based prognostic model for hormone receptor-positive breast cancer using nine-gene expression signature," *World Journal of Oncology*, vol. 14, no. 5, p. 406, 2023.

[186] A. García-Barragán, O. Solarte-Pabón, G. Nedostup, M. Provencio, E. Menasalvas, and V. Robles, "Structuring breast cancer spanish electronic health records using deep learning," in *2023 IEEE 36th international symposium on computer-based medical systems (CBMS)*, IEEE, 2023, pp. 404–409.

[187] K. Poudel, M. Uddin, R. Kommu, S. Muhammed, N. Hasan, and S. Hamdan, "Healthcare text analytics using recent ml techniques," in *International Conference on Advances in Computing Research*, Springer, 2023, pp. 134–142.

[188] L. Zeng et al., "The innovative model based on artificial intelligence algorithms to predict recurrence risk of patients with postoperative breast cancer," *Frontiers in Oncology*, vol. 13, p. 1 117 420, 2023.

[189] M. A. Guvakova and S. Sokol, "The g3mclass is a practical software for multiclass classification on biomarkers," *Scientific Reports*, vol. 12, no. 1, p. 18 742, 2022.

[190] M. K. Doma, K. Padmanandam, S. Tambvekar, K. Kumar, B. Abdualgalil, and R. Thakur, "Artificial intelligence-based breast cancer detection using wpso," *International Journal of Operations Research and Information Systems (IJORIS)*, vol. 13, no. 2, pp. 1–16, 2022.

[191] R. Schiappa et al., "Ruby: Natural language processing of french electronic medical records for breast cancer research," *JCO Clinical Cancer Informatics*, vol. 6, e2100199, 2022.

[192] Z.-H. Ren et al., "Biochemddi: Predicting drug–drug interactions by fusing biochemical and structural information through a self-attention mechanism," *Biology*, vol. 11, no. 5, p. 758, 2022.

[193] D. P. Sahoo, M. Rout, P. K. Mallick, and S. R. Samanta, "Comparative analysis of medical images using transfer learning based deep learning models," in *2022 International Conference on Advancements in Smart, Secure and Intelligent Computing (ASSIC)*, IEEE, 2022, pp. 1–8.

[194] T. H. Ghorpade and S. K. Shinde, "Correlation between image and text from unstructured data using deep learning," in *2022 6th International Conference On Computing, Communication, Control And Automation (ICCUBEA*, IEEE, 2022, pp. 1–6.

[195] O. J. Achilonu, E. Singh, G. Nimako, R. M. Eijkemans, and E. Musenge, "Rule-based information extraction from free-text pathology reports reveals trends in south african female breast cancer molecular subtypes and ki67 expression," *BioMed Research International*, vol. 2022, no. 1, p. 6 157 861, 2022.

[196] N. S. Pagad et al., "Clinical text data categorization and feature extraction using medical-fissure algorithm and neg-seq algorithm," *Computational Intelligence and Neuroscience*, vol. 2022, no. 1, p. 5 759 521, 2022.

[197] E. Qumsiyeh and R. Jayousi, "Biomedical information extraction pipeline to identify disease-gene interactions from pubmed breast cancer literature," in *2021 International Conference on Promising Electronic Technologies (ICPET)*, IEEE, 2021, pp. 1–6.

[198] M. Zhou et al., "Extracting bi-rads features from mammography reports in chinese based on machine learning," *Journal of Flow Visualization and Image Processing*, vol. 28, no. 2, 2021.

[199] P. Symeonidis, G. Manitaras, and M. Zanker, "Accurate and safe drug recommendations based on singular value decomposition," in *2023 IEEE 36th International Symposium on Computer-Based Medical Systems (CBMS)*, IEEE, 2023.

[200] Z. Zheng et al., "Drug package recommendation via interaction-aware graph induction," in *Proceedings of the Web Conference 2021*, 2021.

[201] S. Fouladvand et al., "Graph-based clinical recommender: Predicting specialists procedure orders using graph representation learning," *Journal of Biomedical Informatics*, vol. 143, p. 104 407, 2023, Epub 2023 Jun 3. DOI: 10.1016/j.jbi.2023.104407.

[202] X. Li et al., "Dgcl: Distance-wise and graph contrastive learning for medication recommendation," *Journal of Biomedical Informatics*, vol. 139, p. 104 301, 2023.

[203] S. Zhang, J. Li, H. Zhou, Q. Zhu, S. Zhang, and D. Wang, "Merits: Medication recommendation for chronic disease with irregular time-series," in *2021 IEEE International Conference on Data Mining (ICDM)*, 2021, pp. 1481–1486. DOI: 10.1109/ICDM51629.2021.00192.

[204] S. Kim et al., "Personalized exercise recommender systems for rehabilitation using graph neural networks," *Journal of Korean Institute of Communications and Information Sciences*, vol. 47, no. 4, pp. 644–655, 2022.

[205] P. Symeonidis, T. Kostoulas, V. Danilatou, C. Andras, and S. Chairistanidis, "Mortality prediction and safe drug recommendation for critically-ill patients," in *2022 IEEE 22nd International Conference on Bioinformatics and Bioengineering (BIBE)*, 2022, pp. 79–84. DOI: 10.1109/BIBE55377.2022.00025.

[206] J. G. D. Ochoa and F. E. Mustafa, "Graph neural network modelling as a potentially effective method for predicting and analyzing procedures based on patients' diagnoses," *Artificial Intelligence in Medicine*, vol. 131, p. 102 359, 2022.

[207] X. Lai, "Attention mechanism and graph augmented network for medication recommendation," in *Third International Conference on Biomedical and Intelligent Systems (IC-BIS 2024)*, vol. 13208, SPIE, 2024.

[208] P. Symeonidis, S. Chairistanidis, and M. Zanker, "Deep reinforcement learning for medicine recommendation," in *2022 IEEE 22nd International Conference on Bioinformatics and Bioengineering (BIBE)*, IEEE, 2022.

[209] S. Wang et al., "Order-free medicine combination prediction with graph convolutional reinforcement learning," in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 2019.

[210] Z. Sun et al., "Deep dynamic patient similarity analysis: Model development and validation in icu," *Computer Methods and Programs in Biomedicine*, vol. 225, p. 107 033, 2022.

[211] J. Huo et al., "Mifnet: Multimodal interactive fusion network for medication recommendation," *The Journal of Supercomputing*, vol. 80, no. 9, pp. 12 313–12 345, 2024.

[212] N. M. T. Phan et al., "Fastrx: Exploring fastformer and memory-augmented graph neural networks for personalized medication recommendations," *ACM Transactions on Intelligent Systems and Technology*, vol. 15, no. 6, pp. 1–21, 2024.

[213] X. Li et al., "Rest: Drug-drug interaction prediction via reinforced student-teacher curriculum learning," in *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, 2023.

[214] F. Gong et al., "Smr: Medical knowledge graph embedding for safe medicine recommendation," *Big Data Research*, vol. 23, p. 100 174, 2021.

[215] F. Zhou and S. Uddin, "Fusion of graph and natural language processing in predictive analytics for adverse drug reactions," in *Proceedings of the 2024 Australasian Computer Science Week*, 2024, pp. 65–68.

[216] R. Mishra and S. Shridevi, "Knowledge graph driven medicine recommendation system using graph neural networks on longitudinal medical records," *Scientific Reports*, vol. 14, no. 1, p. 25 449, 2024.

[217] M. Liu, X. Shen, and W. Pan, "Deep reinforcement learning for personalized treatment recommendation," *Statistics in Medicine*, vol. 41, no. 20, pp. 4034–4056, 2022.